NAVSPASUR SYSTEM PERFORMANCE ANALYSIS

Technical Report
for
Contract N00014-87-C-2547

Prepared by:

Dr. E. James Wadiak

Dr. Michael D. Andrews

INTERFEROMETRICS INC.
8150 Leesburg Pike
Vienna, Virginia 22180

16 March 1988

EXECUTIVE SUMMARY

We have applied a system modelling approach to the problem of projecting the performance of an enhanced NAVSPASUR system. We have modelled the performance characteristics of the current system as well as the out-of-plane station (OOPS) and enhanced signal processing options described in the NAVSPASUR System Development Plan, Phase II. The objective of this modelling process was to evaluate the ability of NAVSPASUR to provide satellite position and velocity measurements to the accuracy required for first pass orbit determination (FPOD). The FPOD position and velocity requirements have been taken to be 1.67 km and 2 km/min, respectively, as specified in the NAVSPASUR System Development Plan, Phase I. (RH/mgm)

The major results of this effort, described in detail in this report, are summarized below.

1. The current system appears to be performing near optimal levels in terms of the resultant position and velocity uncertainties for single pass data. For satellites with a 1 meter^2 effective radar cross-section, accurate position determinations are limited to objects crossing the fence over the continental U.S. (CONUS) at altitudes below about 3000 km. Velocity determinations are insufficient to meet FPOD requirements at virtually any fence crossing location. Typical velocity errors range between 3 and 18 km/minute over much of the fence at altitudes between 200 and 2000 km. There is a conspicuous degradation in low altitude system performance to the west of longitude 100^0 W, with velocity errors up to 35 km/min and positional errors in excess of 3 km at altitudes below 200 km. For altitudes greater than 2000 km, velocity errors generally exceed 20 km/min across the full extent of the fence.

2. The improvement of Doppler tracking capability to ±1 Hz and the addition of Doppler rate (chirp) tracking to ±4 Hz/sec results in typical improvements in velocity determination of 30 - 50% over most of the fence, while positional accuracy remains effectively unchanged. The effect of the western "hole" is somewhat diminished, but not entirely eliminated. However, even at this level of performance, the system is incapable of meeting the specified velocity requirements at most fence crossing locations.

3. The addition of three new out-of-plane receiving stations (OOPS) with phase and phase rate measurement

(1)

capability to the current array (assuming ±10 Hz Doppler processing and no chirp measurement) produces two significant improvements over the present system. First, there is an overall improvement of more than an order of magnitude in both position and velocity accuracy for all illuminated objects. Secondly, with a judicious choice of the OOPS locations, the western "hole" in the fence coverage is virtually eliminated. For this system, velocity errors are less than 2 km/min for virtually all CONUS fence crossings at altitudes up to 3000 km. Position errors meet the stated requirement for all orbits investigated, with the exception of 100 km orbits in the far west.

4. If ±10 Hz/sec chirp processing is added to the above system, only marginal improvement is realized. However, if the Doppler and chirp measurement capabilities are enhanced to achieve accuracies of ±1 Hz and ±4 Hz/sec, respectively, an additional factor of 3 improvement is attained. For this case, velocity errors are typically between 0.1 and 0.6 km/min up to altitudes of 4000 km for all CONUS fence crossings, except for 100 km altitudes at the edges of CONUS. Performance up to 3° longitude off the coast is within 1 km/min at altitudes of 500-4000 km.

5. The addition of 3 OOPS which provide only Doppler and chirp measurements could meet the velocity requirements for almost all CONUS fence crossings at altitudes up to 1600 km, assuming Doppler and chirp measurement accuracies of ±10 Hz and ±10 Hz/sec, respectively. If measurement accuracies of ±1 Hz and ±4 Hz/sec can be achieved, velocity errors are reduced to less than 1 km/min up to 4000 km altitudes. For both cases the position errors are within the required accuracy for altitudes up to 4000 km, with the exception of 100 km orbits in the west.

6. The results of our analysis indicate that the target position and velocity requirements can be met with an enhanced NAVSPASUR system which appears to be both technically and economically feasible. However, we have not at this point addressed the specifics of the design, software and hardware requirements, data transfer protocols, or data analysis algorithms which would be required to implement these enhancements.

I. Introduction

We have developed a set of algorithms which are designed to accurately model the current performance characteristics of the NAVSPASUR full-Doppler system both in terms of amplitude response and inherent phase accuracy. We have applied these algorithms to a family of satellite orbits to generate model NAVSPASUR phase and amplitude data sets. These model data were then input into a nonlinear least squares fitting routine used to estimate the likely position and velocity uncertainties as a function of satellite radar cross-section, altitude, longitude at fence crossing, and orbital inclination. In addition to performing this analysis for the current system configuration, we have extrapolated current receiver site performance to proposed additional out-of-plane stations (OOPS), and have evaluated the potential impact such stations would have on overall system performance.

We have also examined the effect of improved Doppler signal processing for both the current system and an expanded system including three OOPS. The impact of adding chirp processing to both the current and expanded systems was examined as well. Finally, we explored the consequences of adding OOPS which provide Doppler and chirp tracking only, with no phase measurement capability.

We describe below the basic assumptions we have adopted for this investigation. The details of the data modelling and error analysis methods are described in Sections II and III, while the analysis of our results is contained in Section IV. In Section V we discuss the areas in which further research is either ongoing or required. Finally, in Section VI we summarize the results of the current research effort and discuss their implications for future changes to the NAVSPASUR system.

a). Approach and Basic Assumptions

The primary objective of this project was to predict the system performance of an enhanced NAVSPASUR system as envisioned in the options presented in the System Development Plan, Phase II. For a system as complex as a multi-station bi-static radar, the net uncertainties in the satellite position and velocity which result from known uncertainties in the measured quantity, antenna phase, depend nonlinearly on both the phase measurement errors and the transmitter-satellite-receiver geometric

relationships. For such a system the most straightforward method of predicting system performance is a modelling approach.

For the current system we have operational data available against which model data can be compared. Given the availability of this data one might expect that the analysis of current system performance should proceed solely on the basis of real data. While we concede the value in such an approach, we have opted to base our analysis on a system model approach for several reasons.

First, it is clear that the real data contain systematic errors which are at present poorly understood and which may be in part correctable. These include both phase calibration errors and clock offset errors, among others. Their inclusion in the current system analysis (by virtue of using real data) but not in the enhanced system analysis would make a comparision between the different systems difficult. Secondly, the necessity to include clock offsets in a least squares solution using real data would double the number of parameters to be solved for and would increase the computational load significantly. Finally, the modelling approach allows us to specify the target size and satellite orbital parameters, so that we can investigate the system performance over the full range of satellite orbits in which we are interested.

Although our formal analysis of the current system performance is limited to modelled data, we have compared our results to the daily aggregate system performance reports compiled at Dahlgren. We find that the results we obtain from our system model are consistent with these summary reports, indicating that the approach we have adopted is basically sound.

In our analysis we have made several assumptions which we enumerate here, deferring a discussion of their significance until a later point.

1.) The current system data acquisition rate (nominally 1/55 second) and phase data quantization factor (1/64 rotation, or $5^\circ\!.6$) would be retained in any future operations as studied in this report.

2). The operation of any additional OOPS will be such that the phase difference measurement errors would be qualitatively the same and of the same RMS magnitude as those for the current in-plane stations.

3). The measurement errors inherent in the determination of the individual antenna array phases are randomly

distributed and vary smoothly with received signal amplitude according to the error model developed in the report *The Determination of NAVSPASUR Phase Errors* (hereafter Paper I). Any systematic errors present in the system are assumed to be either correctable or of sufficiently small magnitude so that their presence does not significantly alter the results we obtain.

4). A Doppler search and tracking accuracy of ±1 Hz is attainable with available hardware using standard signal processing techniques.

II. Data Modelling Technique

The primary objective in generating the model data was to produce data scans which closely resembled, in qualitative and quantitative terms, the real data acquired by the NAVSPASUR system for satellites of similar cross-section and orbits. The steps involved in this process were:

1). Generate a family of satellite orbital elements corresponding to the cases to be investigated.

2). Develop an analytic model for transmitter and receiver beam patterns which produce an acceptable received amplitude response as a function of satellite position.

3). For each data collection time interval, generate the ideal phase differences expected for each independent baseline pair at each receiving station.

4). Introduce random noise on the ideal phase differences using the error model obtained from the phase difference error analysis of Paper I.

5). Calculate the ideal Doppler frequency and chirp for each receiver-transmitter combination.

6). Introduce random errors on the ideal Doppler and chirp values according to the assumed measurement accuracy.

The details of the data modelling algorithm are outlined in the following sections.

a). Satellite orbital elements

We have investigated the system response to satellite passes at altitudes of 100, 200, 300, 500, 700, 1000, 1300, 1600, 2000, 2500, 3000, 3500, and 4000 kilometers which penetrate the fence at longitudes ranging from 75° to 120° west longitude in 5° increments. The full altitude-longitude grid was explored for satellites with an assumed orbital inclination of 85°. This inclination represents a relatively unfavorable case in that these satellites are moving nearly perpendicular to the NAVSPASUR great circle and therefore have a nearly minimum beam crossing time. In order to check the sensitivity of our results to the assumed orbital inclination, we investigated a subset of the altitude-longitude grid at orbital inclinations of 40° and 60°. As expected, the results obtained at these smaller orbital inclinations were marginally better than those at an 85° inclination.

All orbits were assumed to be circular for computational convenience. This restriction is not expected to have any significant effect on our results, at least for objects whose orbital eccentricities are not excessive. However, in the case of satellites with highly elliptic orbits crossing the fence near perigee, some degradation of system performance from the levels predicted herein should be expected, given their relatively higher orbital velocities and correspondingly shorter fence crossing times. Future extensions of this work will address the case of highly elliptic orbits, since some objects of interest do in fact have such orbits.

For each orbit investigated, we calculate the position and velocity of the satellite at the epoch of fence crossing and express these in a rotating, geocentric coordinate system as used by NAVSPASUR. These positions and velocities, together with the assumed radar cross-section, provide the basic input to the data generation algorithms.

b). Transmitter and receiver beam patterns

The transmitter and receiver gains as a function of satellite position were determined from analytic formulae which express the respective antenna gains as a function of north-south and east-west direction angles. The analytic formulae used varied depending on the site.

For each of the transmitter sites, the E-W beam patterns were determined from an 9th order polynomial fit to the beam pattern given in the report *Radiation Pattern Calculation of NAVSPASUR Transmitter Element*, by Dr. Steven L. Berg of Interferometrics, Inc. This beam pattern was obtained from calculations for an inverted-V (arrowhead) dipole antenna mounted horizontally above a finite ground screen of the proper dimensions. This pattern effectively meets the E-W design specifications established for the performance of the transmitter antenna arrays and is believed to accurately represent the current performance of the transmitter sites.

The N-S antenna pattern for the Kickapoo transmitter was modelled as a Gaussian with a full width to half power in the far field of $0^\circ\!.042$, which is the design specification beam width. Since the far field for the Kickapoo antenna array begins at a distance of approximately 15,000 km we have included a term in the N-S beam pattern to compensate for beam broadening due to the target being in the near field of the antenna array. We have compensated for this beam broadening by introducing an additive term to the Gaussian full width to half maximum of $l_{array}/3d_{sat}$ where l_{array} is the overall N-S length of the Kickapoo transmitter array and d_{sat} is the transmitter-satellite distance. The factor of three which appears in the correction term was determined empirically by comparing the model data to real NAVSPASUR data.

The N-S patterns of the two remaining transmitters were modelled as slot antennas of the appropriate N-S length. For all transmitters the gains were normalized so that the integrated power over the upper half-plane was equal to the total radiated transmitter power.

The E-W antenna patterns for the receiver sites were modelled using the analytic formula for a horizontal dipole antenna 0.28 wavelengths above an infinite ground plane of infinite conductivity. For the real receiver arrays, the dipoles are located 0.322 wavelengths above the ground screen. The somewhat lower height we use was chosen to compensate for the beam narrowing due to the finite ground screen. Using our adopted model we obtain an E-W full width to half power which is the same as the design specifications for the receiver arrays.

The receiver N-S patterns were modelled as slot antennas of the appropriate length. The resultant N-S patterns agree well with the design specifications for both the high-altitude and low-altitude stations. The receiver patterns were normalized to produce unity gain integrated over the upper half-plane.

c). Generation of the ideal amplitude and phase data

For each of the test orbits described in Section IIa, we generated a set of model data scans. A separate data scan was generated for each receiver/transmitter combination for which the satellite was at least 2^0 above the horizon at both the receiver and transmitter, and for which the received signal amplitude was -152 dBm or greater for at least one data collection interval (nominally 1/55 sec). The amplitude cutoff of -152 dBm represents a signal-to-noise ratio of approximately 1:1 in the full-Doppler mode, which is a reasonable detection threshold based on an inspection of full-Doppler mode NAVSPASUR data.

For each combination of transmitter/receiver pairs which met the elevation test, we began calculating the expected received amplitude 40 time intervals (nominally 40/55 sec) prior to fence crossing. The amplitudes were calculated based on the satellite position at each time interval, the assumed transmitter and receiver beam patterns, and the assumed effective radar cross section. If no amplitudes greater than the cutoff were found after 110 time interval increments, the calculation was terminated and no data scan was written.

If an amplitude greater than the cutoff is found, we write a data scan header, compute and record the received Doppler frequency (to the nearest 1 Hz) and the data start time, and begin calculating the expected ideal phases. A separate data line is calculated and written for each time interval during which the amplitude remains above the cutoff. Once the received amplitude falls below the cutoff or the number of data lines written equals 55, the data scan is terminated and we proceed to the next transmitter/receiver pair. No secondary scans for the same receiver/transmitter pair are ever generated, even if the signal remains above the cutoff after 55 data lines are written.

For each data line whose amplitude exceeds the cutoff, we calculate the expected antenna phase difference, taken with respect to the designated reference antenna, for each antenna array at the receiving site. This difference is simply the difference in path length from the satellite to the respective antenna arrays, expressed in wavelengths. For a receiver site with n antenna arrays, we generate n-1 phase differences. Implicit in this method is the reasonable assumption that atmospheric and ionospheric effects introduce no additional *differential* path length changes.

Each data line written to the data scan consists of a received amplitude for that time interval and n-1 ideal phase differences. Each data scan consists of between 1 and 55 data

lines for a given transmitter/receiver combination, each with a received amplitude greater than or equal to -152 dBm.

d). The addition of errors to the data

Once an ideal data scan has been completed, we introduce errors onto the ideal phase differences to simulate the errors expected during normal system operation. The phase difference error model we have adopted for this work is the error model produced by Interferometrics, Inc., which is described in Paper I. Using this model we obtain errors which are randomly distributed, are zero in the mean, and whose RMS magnitude is a function of the received signal amplitude.

For each phase difference datum we use the adopted error model to generate a random error based on the received amplitude associated with that datum and add this error to the ideal phase. This process is repeated for all the data points in the scan.

We do not introduce errors to the calculated Doppler frequency and chirp at the time the data scans are written. We defer adding errors to the Doppler data until the least squares fitting step, allowing us to use the same input data set for all Doppler accuracy cases. When the Doppler errors are added, they are random errors with zero mean, as in the case of the phase errors previously discussed.

e). OOPS location and performance specifications

A major objective of this work was to estimate the performance improvements that could be realized by the addition of receiver stations located out of the NAVSPASUR great circle plane. In the current work we have considered an enhanced system which includes three OOPS. We believe this to be the minimum number of additional stations necessary to meet the target position and velocity accuracies from coast to coast. We base this conclusion on a preliminary aperture systhesis analysis using standard radio astronomical techniques, from which we obtained the nominal positions for the three OOPS considered in this study. The geographic locations of the three OOPS are depicted on the map given in Figure 1, and are as follows:

	N. Latitude	W. Longitude
Northeast OOPS	35.09	92.59
Central OOPS	42.55	97.54
Northwest OOPS	37.17	103.38

The results described in this report detail the type of performance enchancement that can be obtained with the addition of multiple OOPS stations. However, we do not believe, nor do we represent, that the configuration used herein is the only, or even the optimal, configuration for an OOPS enhancement. We are continuing to study the range of possible station locations, as well as the operational performance requirements for the OOPS. The results of these further studies will be provided when available.

The results we derive in this paper are based on the assumption that each OOPS antenna array could produce an effective antenna gain of 35 dB in the direction of the satellite. Preliminary investigations indicate that similar accuracies are obtainable with lower-gain OOPS, with the tradeoff that FPOD capability would be limited to lower-altitude orbits. Since the question of required gain is intimately related to the number and location of the OOPS, we have deferred a detailed analysis of the gain requirements until a later report, at which time we will address the conceptual design of the OOPS.

For the models which assume OOPS with phase measurement capability, the OOPS stations consist of five antenna arrays arranged in a filled-cross configuration with baselines aligned in the N-S and E-W directions. The spacing between antenna arrays in both directions is taken to be 5 km.

III. Least Squares Analysis of System Performance

The expected errors in the derived position and velocity for a given satellite orbit are estimated using a recursive nonlinear least squares fitting and error estimation routine with data weighting, similar to that described in Paper I. While the least squares technique is not a viable algorithm for routine processing of real NAVSPASUR data (due to the requirement that one have a good *a priori* estimate of the satellite position and velocity), in the case of model data, where the *a priori* position is known, it does provide a valid means to estimate the accuracy which can be achieved by the system. For the present application, the least squares fitting is applied to all data scans for a particular satellite pass simultaneously - *i.e.*, all available combinations of receiving and transmitting sites are used to obtain estimates for the satellite position and velocity and the likely error in these estimates.

The least squares routine yields estimates for six parameters - three position and three velocity coordinates at the

epoch of fence crossing, from which the satellite orbital elements can be obtained. Since we deal here with modelled as opposed to real data, we do not need to solve for any additional parameters such as propagation effects or relative errors in the clock times between the different receiving sites. In fact, the data generation and least squares fitting programs are set up to do all calculations in "satellite" time, so that light travel times are explicitly corrected for in the data modelling and fitting routines by treating them identically in both cases.

The coordinate system which we adopt in the multi-station least squares fitting process is based on the NAVSPASUR great circle, so that the x-y coordinate plane is coincident with the NAVSPASUR great circle plane and the z axis is normal to the great circle plane. Thus the coordinate system is geocentric and rotates with the rotating earth. This choice of coordinate system allows us to determine the position and velocity errors in a coordinate system which is related to the special geometry of the NAVSPASUR system. In this coordinate system the relative magnitudes of the in-plane and plane-normal errors provide insight into the limitations imposed by the current geometry as well as indicating the favored geometry for any system improvements. The diagram shown in Figure 2 shows the relative orientation of the NAVSPASUR great circle coordinate system in the rotating geocentric reference frame.

In addition to generating phase data, we also calculate the expected Doppler and chirp for each transmitter-receiver combination. We have modified the least squares program to make use of this information in obtaining a best estimate of position and velocity. When Doppler and chirp data are included in a given test case, we introduce random errors on their nominal values according to their assumed accuracy. As with the phase data, these data are appropriately weighted by their RMS error in the least squares fitting routine.

Once the least squares solutions to the satellite position and velocity have been obtained, we also have available the error matrix F^{-1} (see Paper I). The diagonal terms of the error matrix are the variances for each of the parameters, while the off-diagonal elements are the covariance terms, which give the correlations between the errors in the parameters. From the error matrix we can calculate the expected error in each component of the position and velocity, as well as the expected total (3-dimensional) error for each.

IV. Analysis of Results

The multi-station least squares data fitting procedure outlined in the previous section has been applied to the simulated data scans described in Section II. We have obtained estimates of the position and velocity uncertainties as a function of fence crossing longitude and altitude for each of following system configurations:

1). Current receiver configuration (six sites) with phase and Doppler measurements at all sites.

2). Current configuration with phase, Doppler, and chirp measurements at all sites.

3). Current and 3 OOPS locations with phase and Doppler measurements at all sites.

4). Current and 3 OOPS locations with phase, Doppler and chirp measurements at all sites.

5). Current and 3 OOPS locations with phase measurements at current sites only, Doppler at all sites.

6). Current and 3 OOPS locations with phase measurements at current sites only, Doppler and chirp at all sites.

For each of these configurations we have obtained predictions of the system response to 130 test orbits which sample the range of altitudes and fence crossing longitudes enumerated in Section IIa. The assumed orbital inclination was 85^0 and the assumed radar cross-section was 1 meter^2. In general, the predicted amplitudes for orbits above 4000 km were such that little or no useful data was generated, so we exclude these orbits from further consideration.

With each of the above configurations we have run the full grid of model orbits for different Doppler and, where included, chirp accuracies. The range of Doppler measurement accuracy investigated was ±1 Hz to ±20 Hz, while the chirp accuracy was varied between ±4 Hz/sec and ±20 Hz/sec.

The quantitative results from our model runs are summarized in Tables 1a-12d. These Tables list for each case the expected total (3-dimensional) position and velocity errors as a function of fence crossing position, and provide a breakdown of the velocity errors into NAVSPASUR great circle in-plane (2-D) and plane-normal components. It is clear from an examination of

these Tables that the measurement accuracy of the NAVSPASUR system for any given configuration cannot be characterized by a single number, as the expected error depends sensitively on the longitude and altitude of fence crossing. In general, all cases studied exhibit their best performance for fence crossing longitudes between 85°W and 100°W, at altitudes between 200 and 1600 km. This is a result of the higher density of receiver stations to the east of the Kickapoo transmitter site.

We provide in the following sections a detailed analysis of the system performance for the various options examined.

a). Options utilizing the current receiver configuration

The first case studied corresponds to the current capability model as described in Table 5-3 of the System Development Plan, Phase II (hereafter SDP). For this case we assume a Doppler measurement accuracy of ±20 Hz with no chirp. In discussions with the staff at NAVSPASUR headquarters we have been advised that the phase-fitting method of obtaining the Doppler frequency yields a typical accuracy of ±3-4 Hz. However, for consistency with the SDP, we have run a ±20 Hz model. The results of this model are summarized in Tables 1a-d.

An examination of Table 1a indicates that, with the exception of low altitude satellites in the west, the present system meets the position determination objective of ±1.67 km over CONUS up to an altitude of 3000 km. The velocity performance of the current system, however, does not meet the ±2 km/min requirement at any fence crossing locations with the exception of low altitude satellites at 90° west longitude (see Table 1b). A comparison of in-plane vs. plane-normal velocity performance, listed in Tables 1c and 1d respectively, demonstrates that the velocity errors are dominated by the uncertainties in the plane-normal component. An analysis of the Doppler measurement equations for the NAVSPASUR system shows that, due to the peculiar geometry of the current system, the received Doppler frequency is insensitive to the plane-normal component of the satellite velocity, so that improved Doppler tracking accuracy from in-plane receivers will not substantially affect the overall velocity accuracy. We include a full analysis of the Doppler measurement equations in Appendix A, to which the interested reader is referred.

The assesment that improved Doppler accuracy will not improve overall velocity accuracy is borne out by the results of our second model, which was based on the current receiver configuration with an assumed a Doppler accuracy of ±4 Hz. While

the improved Doppler accuracy yields in-plane velocity measurements which are a factor of 5 better than the first case (compare Tables 1c and 2c), there is no net improvement in the plane-normal velocity error (Tables 1d and 2d) or in the resulting total velocity error (Tables 1b and 2b).

Although the received Doppler frequency is insensitive to plane-normal velocity changes, the chirp does contain some information on plane-normal velocity (see Appendix A for details). We have therefore investigated the effect of utilizing chirp information in the fitting process. For the current receiver configuration, we find that a chirp measurement accuracy of ±20 Hz/sec combined with a Doppler measurement accuracy of ±20 Hz yields a typical velocity improvement of 2-4% as compared to the ±20 Hz Doppler measurement alone (ref. Table 3b). Clearly, if chirp is to yield a significant improvement to the velocity measurement error, chirp measurement accuracies much better than ±20 Hz/sec are required. As an example, if the chirp measurement accuracy is improved to ±4 Hz/sec, together with a Doppler measurement accuracy of ±1 Hz, velocity errors are reduced by 30-50% or more (see Tables 4b,d). This level of Doppler and chirp measurement accuracy represents a reasonable limit to what can be achieved with the current receiver system, based on our fits to NAVSPASUR phase4 data.

It is apparent from Table 4b that, even with ±1 Hz Doppler and ±4 Hz/sec chirp capability, the current system cannot achieve the target requirement of ±2 km/min velocity error over most of the fence. This limitation results from the planar geometry of the current system. In order to correct this deficiency, one must either view the satellite from an off-plane location while it lies in the NAVSPASUR great circle plane, or illuminate the satellite while it lies outside the plane.

b). Configurations including out-of-plane stations

We next investigated the effect of adding three out-of-plane receiving stations to the current system. As we have discussed in Section IIe, we believe this to be the minimum number of OOPS required to give adequate coast-to-coast performance. Two different types of OOPS were considered. In one set of cases we considered the effects of adding OOPS which provided both phase and Doppler data, while in the other case we examined the effect of limiting the OOPS to Doppler measurements alone. For both cases we also considered the impact of adding chirp measurement capability.

i. Doppler-only OOPS

In general, Doppler measurements made from receiving stations well removed from the NAVSPASUR great circle plane contain significant information concerning the plane-normal velocity (see Appendix A for details). For systems employing suitably located OOPS receivers, we expect the Doppler-derived plane-normal velocity errors to be comparable to the in-plane errors, and we further expect the improvement in velocity determination to be proportional to the improvement in Doppler measurement accuracy.

Our models indicate that the addition of three Doppler-only OOPS as described in Section IIe with ±20 Hz Doppler accuracy would yield up to a factor of three improvement in velocity performance over the current configuration with the same Doppler accuracy (ref. Table 5b). As expected, this improvement comes primarily from an enhanced plane-normal velocity determination capability, as seen from a comparison of Table 5c with Table 1c and Table 5d with Table 1d. While the improvement is significant, however, the velocity errors still exceed the target requirement over much of the fence.

If the Doppler accuracy is improved to ±10 Hz the velocity errors are reduced by an additional factor of ~2, resulting in errors that fall within the target requirement over most of the fence at altitudes up to 1300 km (see Table 6b).

In contrast with the results obtained for the system based on the current six receivers, the addition of chirp measurement capability to the OOPS configurations yields no significant improvement in system performance. We show in Tables 7a-d the results we obtain for an OOPS configuration with ±10 Hz Doppler accuracy and ±10 Hz/sec chirp accuracy. These results are almost identical to those in Tables 6a-d, where no chirp capability was assumed.

The final Doppler-only OOPS configuration we investigated assumed a Doppler accuracy of ±1 Hz and a chirp accuracy of ±4 Hz/sec at all stations. This configuration would produce velocity uncertainties well within the target requirement at virtually all fence crossing locations up to an altitude of 4000 km (see Tables 8a-d). We should note here that for this configuration the positional errors do exceed the target requirement at the edges of CONUS at altitudes above 3000 km.

ii. Phase and Doppler OOPS

The measurement of phase offsets between spatially separated antenna arrays at the OOPS locations can provide potentially dramatic improvement in system performance. For a phase-capable OOPS site as described in Section IIe with Doppler accuracy of ±20 HZ, we obtain velocity errors within the requirement over most of CONUS up to 2500 km altitude (Table 9b). The in-plane and plane-normal velocity errors for this configuration are roughly equal (Tables 9c,d). Moreover, the position errors are vastly improved over the systems previously discussed (Table 9a), with errors of 100 meters or less over most of the fence up to 4000 km altitude.

Improving the Doppler accuracy to ±10 Hz nets an additional velocity improvement of 5-15% over most of the fence (Tables 10a-d), effectively extending the altitude limit for meeting the velocity requirement to 3000 km. Again, we find that adding chirp capability produces no significant gains, with the exception of some marginal improvement for low altitude orbits near the edges of CONUS (Tables 11a-d).

Finally, we analyzed the system performance for an assumed Doppler accuracy of ±1 Hz and a chirp accuracy of ±4 Hz/sec. The results, shown in Tables 12a-d, indicate that such a system would be capable of achieving the target position and velocity requirements up to an altitude of 4000 km over all of CONUS with the exception of 100 km orbits at the extreme edges of CONUS. Further, this system could provide off-coast (up to 5^{0} longitude) performance within the target requirements for orbits between 500 and 4000 km. Typical velocity errors over most of the fence are less than 0.5 km/min, while typical position errors are less than 100 meters.

V. Identification of Further Work

Further research is either currently underway or planned in several areas related to the work described in this report. The highest priority research effort is to explore the possible range of number, location, and operational design specifications for the OOPS. The current work demonstrates that the required position and velocity accuracies can be obtained with 3 OOPS with the performance characteristics assumed in this report. However, a detailed investigation is required into the tradeoffs between the number of OOPS sites and their design and location in order to optimize system performance in a cost effective manner. In

support of this optimization process we are instituting several enhancements to the system modelling algorithms.

As we have have shown in this report, the uncertainties in the satellite velocity depend sensitively on the assumed Doppler measurement accuracy at the OOPS locations. We are therefore analyzing current phase4 data to determine the level of Doppler and chirp accuracy which can be attained with the present receiver setup. This analysis will allow us to generate a more realistic Doppler/chirp error model, where the expected errors in these parameters depend on the received signal-to-noise ratio and the number of useable data lines.

We are also working to develop a realistic OOPS gain model, based on antenna pattern calculations for a two-dimensional phased array of multi-element dipoles. These calculations will allow us to specify the OOPS gain as a function of azimuth and elevation to the target satellite as opposed to a constant-gain model as assumed for this report.

When the above enhancements are incorporated into the sytem model, we will proceed with the determination of the optimum system configuration and the development of design specifications for the OOPS receivers.

In addition to the above-mentioned efforts, we recognize the need to extend this work to include performance estimates for highly elliptic satellite orbits, and to investigate the performance of both the half-Doppler and quarter-Doppler modes.

V. Summary

We have developed a set of algorithms which allow us to accurately model the full-Doppler mode performance of the current NAVSPASUR system, as well as the proposed enhancements outlined in the NAVSPASUR System Development Plan, Phase II. The enhancements which have been studied to date fall into two categories: 1) systems based on the current six receiver sites but requiring additional and/or improved signal processing, and 2) systems requiring the addition of three new out-of-plane receiving stations, either with or without enhanced signal processing capability.

The results of our analysis indicate that none of the enhancements which are based solely on the current six receiving stations are capable of meeting the stated velocity accuracy goal of ±2 km/min over all of CONUS. With the addition of three OOPS with both phase and Doppler/chirp measurement capability, it is

possible to meet the velocity accuracy requirement over CONUS up to an altitude of 4000 km. If the three OOPS are configured for Doppler/chirp measurement only, velocity errors within the stated specification may still be possible over CONUS up to 4000 km if Doppler/chirp measurement accuracies of ±1Hz and ±4Hz/sec, respectively, can be achieved.

We have demonstrated in this work that there exists at least one system enhancement geometry which is capable of providing single-pass position and velocity accuracies meeting the target requirements specified in the NAVSPASUR System Development Plan, Phase I. The results presented herein indicate that these requirements can be met with an expanded system which would include three OOPS receiver sites utilizing the current transmitters for satellite illumination. While our results show that the accuracy requirements can be met with the system enhancements proposed herein, additional work must be done to fully explore the range of possible design and location options for the OOPS. We are currently investigating the technical issues relating to the OOPS design and location, and will present our findings in an upcoming report.

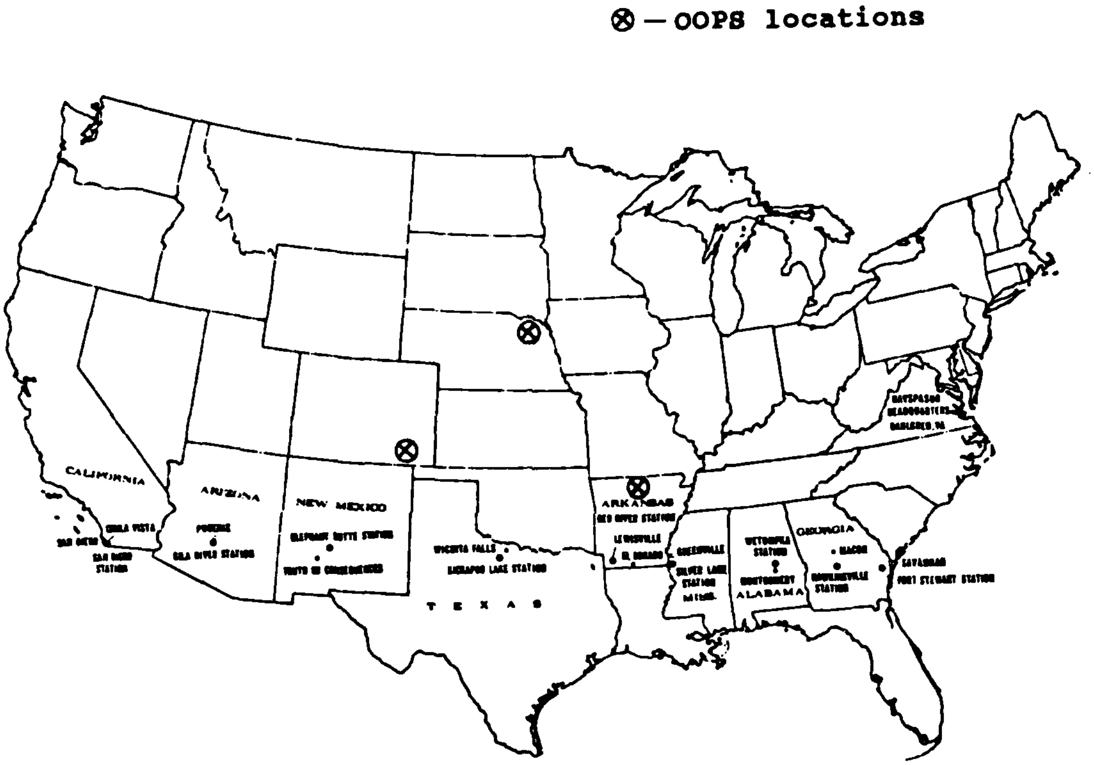

FIGURE 1: Map depicting the geographic locations of the current NAVSPASUR transmitter and receiver stations, as well as the nominal positions of the three OOPS receiver sites used in this work.

TABLE 1a

CURRENT CONFIGURATION DOPPLER +- 20 HZ NO CHIRP

RMS TOTAL POSITION ERRORS IN KM

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	55.56	108.0	1.46	0.65	0.47	0.50	0.60	0.84	1.35	2.81	X	X
	115	5.97	0.48	0.19	0.10	0.09	0.13	0.19	0.29	0.46	0.85	1.58	4.73	X
W	110	3.26	0.36	0.16	0.09	0.09	0.11	0.15	0.19	0.28	0.48	1.14	1.98	2.68
	105	16.59	3.67	0.46	0.13	0.09	0.09	0.11	0.15	0.21	0.32	0.51	0.98	1.32
L	100	1.30	0.34	0.18	0.10	0.08	0.09	0.10	0.13	0.17	0.24	0.35	0.55	0.83
O	95	0.15	0.06	0.05	0.07	0.09	0.11	0.12	0.15	0.18	0.25	0.37	0.58	0.87
N	90	0.16	0.05	0.04	0.05	0.06	0.08	0.11	0.16	0.23	0.36	0.55	0.82	1.19
G	85	0.33	0.08	0.05	0.04	0.05	0.07	0.12	0.17	0.28	0.47	0.76	1.23	1.75
	80	0.90	0.26	0.21	0.19	0.21	0.23	0.27	0.34	0.48	0.75	1.35	2.23	3.02
	75	X	X	X	207.5	2.29	2.14	2.54	3.33	4.79	8.54	X	X	X

TABLE 1b

CURRENT CONFIGURATION DOPPLER +- 20 HZ NO CHIRP

RMS TOTAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	64.84	121.5	15.16	16.61	20.23	23.29	24.88	27.90	32.29	39.22	137.5	143.1
	115	22.09	17.82	8.42	9.30	10.69	12.55	14.20	16.12	19.23	24.40	31.17	43.12	X
W	110	35.88	16.22	13.36	11.01	10.45	10.83	11.77	13.40	16.71	22.58	30.30	40.63	57.42
	105	36.63	18.47	14.43	10.39	9.42	9.90	11.18	12.84	15.74	20.33	26.05	34.09	42.72
L	100	13.84	9.23	8.03	7.12	7.07	7.52	8.51	9.74	11.70	14.88	19.17	25.50	32.55
O	95	4.12	3.07	3.02	3.54	4.43	6.32	8.08	9.90	12.20	15.74	20.58	27.49	34.68
N	90	2.20	2.01	2.11	2.68	3.55	5.29	7.27	9.45	13.26	18.79	26.43	34.76	45.68
G	85	19.66	7.06	5.31	4.67	4.91	6.16	8.02	9.78	13.04	18.60	26.98	37.38	52.68
	80	13.41	10.58	11.18	9.02	9.12	11.02	12.75	14.38	17.22	22.39	29.56	39.67	54.44
	75	X	X	X	X	33.77	28.15	28.04	41.76	46.46	54.56	334.9	751.3	466.5

TABLE 1c

CURRENT CONFIGURATION DOPPLER +- 20 HZ NO CHIRP

RMS IN-PLANE VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	25.18	39.27	3.43	3.41	3.58	6.48	7.08	8.87	10.40	11.96	73.73	94.79
	115	2.51	2.21	1.61	1.79	2.00	2.34	1.93	2.52	2.75	3.83	4.56	8.38	369.9
W	110	2.95	1.12	1.06	1.19	1.00	1.05	1.17	1.79	2.01	3.01	3.77	6.71	7.26
	105	3.26	1.26	1.01	0.97	1.00	1.11	1.22	1.34	2.07	2.53	2.85	5.41	5.94
L	100	2.63	1.97	0.89	0.82	0.83	0.89	0.98	1.08	1.69	1.92	2.17	2.68	4.86
O	95	2.19	0.97	0.86	0.80	0.69	0.76	1.03	1.59	1.80	2.07	3.19	4.40	5.15
N	90	1.11	0.86	0.74	0.69	0.77	1.10	1.43	1.62	2.81	3.27	3.75	5.30	5.90
G	85	1.53	1.13	1.02	1.13	1.10	1.27	1.80	2.02	2.32	3.87	4.39	6.17	6.80
	80	7.49	6.42	7.03	4.94	2.75	2.32	2.50	2.69	4.16	4.68	5.34	7.29	7.93
	75	X	X	X	X	16.16	13.39	12.28	31.21	35.81	42.36	123.1	194.9	150.9

TABLE 1d

CURRENT CONFIGURATION DOPPLER +- 20 HZ NO CHIRP

RMS PLANE-NORMAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	59.53	87.80	14.61	16.16	19.84	22.32	23.77	26.38	30.36	36.90	84.53	102.9
	115	21.85	17.56	8.20	9.07	10.45	12.28	14.04	15.84	18.98	24.00	30.79	42.04	821.8
W	110	35.29	16.07	13.24	10.89	10.36	10.75	11.68	13.24	16.56	22.33	30.01	39.92	56.76
	105	36.30	18.33	14.33	10.30	9.34	9.81	11.09	12.75	15.58	20.14	25.87	33.59	42.21
L	100	13.25	8.91	7.91	7.03	6.98	7.43	8.41	9.64	11.53	14.71	19.02	25.30	32.13
O	95	3.42	2.88	2.86	3.40	4.33	6.23	7.97	9.69	12.00	15.53	20.31	27.09	34.25
N	90	1.69	1.66	1.88	2.51	3.39	5.10	7.05	9.24	12.85	18.44	26.12	34.30	45.24
G	85	19.47	6.92	5.17	4.48	4.72	5.96	7.72	9.48	12.77	18.10	26.59	36.80	52.13
	80	10.69	8.19	8.49	7.42	8.59	10.70	12.40	14.05	16.58	21.80	29.05	38.87	53.67
	75	X	X	X	120.3	29.55	24.66	25.16	27.38	29.04	33.36	243.3	629.4	376.4

TABLE 2a

CURRENT CONFIGURATION DOPPLER +- 4 HZ NO CHIRP

RMS TOTAL POSITION ERRORS IN KM

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	55.56	108.0	1.46	0.65	0.47	0.50	0.60	0.84	1.35	2.81	X	X
	115	5.97	0.48	0.19	0.10	0.09	0.13	0.19	0.29	0.46	0.85	1.58	4.73	X
W	110	3.26	0.35	0.16	0.09	0.09	0.11	0.15	0.19	0.28	0.48	1.14	1.98	2.68
	105	16.59	3.67	0.46	0.13	0.09	0.09	0.11	0.15	0.21	0.32	0.51	0.98	1.32
L	100	1.30	0.34	0.18	0.10	0.08	0.09	0.10	0.13	0.17	0.24	0.35	0.55	0.83
O	95	0.15	0.06	0.05	0.07	0.09	0.11	0.12	0.15	0.18	0.25	0.37	0.58	0.87
N	90	0.16	0.05	0.04	0.05	0.06	0.08	0.11	0.16	0.23	0.36	0.55	0.82	1.19
G	85	0.33	0.08	0.05	0.04	0.05	0.07	0.12	0.17	0.28	0.47	0.76	1.23	1.75
	80	0.90	0.26	0.21	0.19	0.21	0.23	0.27	0.34	0.47	0.75	1.35	2.23	3.02
	75	X	X	X	207.5	2.27	2.12	2.53	3.29	4.75	8.50	X	X	X

TABLE 2b

CURRENT CONFIGURATION DOPPLER +- 4 HZ NO CHIRP

RMS TOTAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	64.83	121.5	14.78	16.27	19.92	22.40	23.81	26.52	30.65	37.44	137.4	142.3
	115	21.96	17.68	8.23	9.12	10.50	12.33	13.87	15.85	18.89	23.83	30.52	42.33	X
W	110	35.77	16.16	13.31	10.94	10.35	10.74	11.62	13.27	16.54	22.21	29.82	40.09	56.98
	105	36.49	18.29	14.33	10.26	9.29	9.80	11.08	12.72	15.54	20.08	25.57	33.68	42.32
L	100	13.36	9.02	7.78	7.00	6.98	7.44	8.42	9.65	11.53	14.70	18.91	24.93	32.20
O	95	3.56	2.92	2.89	3.42	4.34	6.23	7.97	9.70	12.00	15.50	20.27	26.77	34.31
N	90	1.72	1.67	1.89	2.5[illegible]	3.40	5.10	7.07	9.27	12.87	18.44	25.87	34.37	45.31
G	85	19.33	6.96	5.21	4.51	4.75	5.98	7.75	9.51	12.78	18.09	26.29	36.89	52.26
	80	11.30	8.61	8.91	7.6[illegible]	8.68	10.75	12.46	14.06	16.62	21.76	28.59	39.02	53.88
	75	X	X	X	X	29.47	2[illegible]75	25.24	28.89	31.10	36.32	334.9	749.8	466.9

TABLE 2c

CURRENT CONFIGURATION DOPPLER +- 4 HZ NO CHIRP

RMS IN-PLANE VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	25.17	39.27	0.71	0.70	0.72	1.31	1.43	1.79	2.11	2.45	73.67	94.42
	115	0.67	0.45	0.33	0.36	0.40	0.47	0.39	0.51	0.55	0.77	0.91	1.70	370.8
W	110	0.83	0.25	0.22	0.24	0.20	0.21	0.23	0.36	0.40	0.61	0.76	1.35	1.47
	105	0.69	0.27	0.21	0.20	0.20	0.22	0.25	0.27	0.42	0.51	0.57	1.09	1.20
L	100	0.63	0.41	0.18	0.17	0.17	0.18	0.20	0.22	0.34	0.39	0.44	0.54	0.98
O	95	0.47	0.20	0.18	0.16	0.14	0.15	0.21	0.32	0.36	0.42	0.64	0.88	1.03
N	90	0.23	0.18	0.15	0.14	0.16	0.22	0.29	0.33	0.57	0.66	0.75	1.06	1.18
G	85	0.33	0.23	0.21	0.23	0.22	0.26	0.36	0.41	0.47	0.78	0.88	1.24	1.37
	80	1.67	1.50	1.67	1.08	0.57	0.47	0.51	0.54	0.84	0.94	1.07	1.47	1.60
	75	X	X	X	X	3.29	2.74	2.50	7.07	8.31	9.94	123.1	194.6	151.0

TABLE 2d

CURRENT CONFIGURATION DOPPLER +- 4 HZ NO CHIRP

RMS PLANE-NORMAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	59.53	87.80	14.60	16.16	19.84	22.31	23.68	26.38	30.36	36.90	84.43	102.2
	115	21.85	17.56	8.16	9.06	10.44	12.27	13.83	15.76	18.84	23.73	30.47	42.04	823.0
W	110	35.29	16.05	13.23	10.88	10.30	10.71	11.59	13.22	16.51	22.15	29.76	39.92	56.76
	105	36.30	18.19	14.26	10.22	9.26	9.76	11.05	12.69	15.49	20.05	25.54	33.59	42.21
L	100	13.01	8.89	7.71	6.96	6.95	7.41	8.39	9.62	11.50	14.67	18.89	24.87	32.13
O	95	3.42	2.87	2.85	3.39	4.31	6.21	7.94	9.66	11.96	15.46	20.20	26.71	34.25
N	90	1.66	1.64	1.86	2.50	3.38	5.08	7.04	9.23	12.83	18.39	25.81	34.30	45.24
G	85	19.20	6.91	5.17	4.47	4.71	5.95	7.72	9.47	12.74	18.04	26.25	36.80	52.13
	80	10.69	8.19	8.49	7.41	8.58	10.69	12.38	14.02	16.54	21.69	28.54	38.87	53.67
	75	X	X	X	120.3	29.21	24.52	25.07	27.38	29.03	33.35	243.3	628.3	376.7

TABLE 3a

CURRENT CONFIGURATION DOPPLER +- 20 HZ CHIRP +- 20 HZ/SEC

RMS TOTAL POSITION ERRORS IN KM

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	48.19	61.94	1.46	0.65	0.47	0.50	0.60	0.84	1.35	2.81	X	X
	115	5.58	0.48	0.19	0.10	0.09	0.13	0.19	0.29	0.46	0.85	1.58	4.73	X
W	110	3.21	0.35	0.16	0.09	0.09	0.11	0.15	0.19	0.28	0.48	1.14	1.98	2.68
	105	14.02	3.62	0.46	0.13	0.09	0.09	0.11	0.15	0.21	0.32	0.51	0.98	1.32
L	100	1.29	0.34	0.18	0.10	0.08	0.09	0.10	0.13	0.17	0.24	0.35	0.55	0.83
O	95	0.15	0.06	0.05	0.07	0.09	0.11	0.12	0.15	0.18	0.25	0.37	0.58	0.87
N	90	0.16	0.05	0.04	0.05	0.06	0.08	0.11	0.16	0.23	0.36	0.55	0.82	1.19
G	85	0.31	0.07	0.04	0.04	0.05	0.07	0.12	0.17	0.28	0.47	0.76	1.23	1.75
	80	0.90	0.26	0.21	0.19	0.21	0.23	0.27	0.34	0.47	0.75	1.35	2.23	3.02
	75	X	X	X	2.12	2.29	2.14	2.54	3.33	4.79	8.54	X	X	X

TABLE 3b

CURRENT CONFIGURATION DOPPLER +- 20 HZ CHIRP +- 20 HZ/SEC

RMS TOTAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	63.68	67.62	13.41	14.95	18.30	22.04	23.78	27.25	31.60	38.33	388.4	275.3
	115	10.75	11.28	7.39	8.53	9.96	11.87	13.49	15.52	18.52	23.69	30.34	42.14	539.0
W	110	10.73	9.50	9.72	9.60	9.52	10.16	11.19	12.91	16.07	21.77	29.45	39.68	55.41
	105	15.81	11.36	10.43	8.92	8.56	9.36	10.66	12.29	15.19	19.66	25.09	33.48	41.81
L	100	7.35	7.24	6.59	6.43	6.63	7.21	8.22	9.48	11.44	14.54	18.69	24.82	32.13
O	95	3.91	2.95	2.93	3.44	4.30	6.09	7.84	9.66	11.91	15.34	20.08	26.81	34.18
N	90	2.16	1.98	2.08	2.64	3.48	5.17	7.14	9.25	12.97	18.26	25.41	34.11	44.58
G	85	5.67	5.16	4.67	4.45	4.75	5.97	7.87	9.58	12.73	18.15	26.01	36.65	51.13
	80	11.22	9.79	10.54	8.62	8.67	10.46	12.27	13.88	16.79	21.72	28.67	38.90	52.92
	75	X	X	X	98.09	29.71	26.15	26.52	41.27	46.09	54.21	263.3	307.1	355.1

TABLE 3c

CURRENT CONFIGURATION DOPPLER +- 20 HZ CHIRP +- 20 HZ/SEC

RMS IN-PLANE VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	39.88	32.32	3.43	3.41	3.58	6.48	7.08	8.87	10.40	11.96	168.6	147.1
	115	2.51	2.21	1.61	1.79	2.00	2.34	1.93	2.52	2.75	3.83	4.56	8.38	243.5
W	110	2.92	1.11	1.06	1.19	1.00	1.05	1.17	1.79	2.01	3.01	3.77	6.71	7.26
	105	3.26	1.26	1.01	0.97	1.00	1.11	1.22	1.34	2.07	2.53	2.85	5.41	5.94
L	100	2.61	1.97	0.89	0.82	0.83	0.89	0.98	1.08	1.69	1.92	2.17	2.68	4.86
O	95	2.19	0.97	0.86	0.80	0.69	0.76	1.03	1.59	1.80	2.07	3.19	4.40	5.15
N	90	1.11	0.86	0.74	0.69	0.77	1.10	1.43	1.62	2.81	3.27	3.75	5.30	5.90
G	85	1.53	1.13	1.02	1.13	1.10	1.27	1.80	2.02	2.32	3.87	4.39	6.16	6.80
	80	7.47	6.41	7.02	4.94	2.75	2.32	2.50	2.69	4.16	4.68	5.34	7.29	7.93
	75	X	X	X	86.99	16.15	13.39	12.28	31.19	35.79	42.33	111.0	122.7	187.4

TABLE 3d

CURRENT CONFIGURATION DOPPLER +- 20 HZ CHIRP +- 20 HZ/SEC

RMS PLANE-NORMAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	43.85	45.74	12.85	14.48	17.89	21.02	22.60	25.67	29.63	35.96	263.0	223.3
	115	10.40	10.98	7.15	8.29	9.71	11.59	13.32	15.24	18.27	23.29	29.96	41.05	369.2
W	110	10.19	9.37	9.61	9.48	9.43	10.08	11.10	12.75	15.92	21.52	29.15	38.97	54.74
	105	15.43	11.23	10.33	8.84	8.48	9.26	10.57	12.19	15.02	19.46	24.90	32.96	41.30
L	100	6.71	6.88	6.48	6.34	6.55	7.12	8.13	9.38	11.27	14.37	18.53	24.61	31.71
O	95	3.18	2.76	2.77	3.31	4.20	6.00	7.72	9.45	11.71	15.14	19.80	26.40	33.74
N	90	1.64	1.63	1.85	2.47	3.32	4.97	6.92	9.04	12.56	17.90	25.10	33.64	44.13
G	85	5.43	5.00	4.52	4.25	4.56	5.77	7.57	9.28	12.46	17.64	25.61	36.06	50.58
	80	8.20	7.26	7.75	6.98	8.13	10.14	11.92	13.55	16.14	21.13	28.14	38.10	52.15
	75	X	X	X	44.32	24.65	22.40	23.38	26.33	28.14	32.44	188.6	263.5	270.2

TABLE 4a

CURRENT CONFIGURATION DOPPLER +- 1 HZ CHIRP +- 4 HZ/SEC

RMS TOTAL POSITION ERRORS IN KM

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	36.11	38.96	1.41	0.64	0.46	0.50	0.60	0.84	1.35	2.81	X	X
	115	4.63	0.48	0.18	0.09	0.09	0.13	0.19	0.29	0.46	0.85	1.58	4.73	X
W	110	3.08	0.35	0.15	0.09	0.09	0.11	0.15	0.19	0.28	0.48	1.14	1.98	2.68
	105	12.74	3.47	0.46	0.13	0.09	0.09	0.11	0.15	0.21	0.32	0.51	0.98	1.32
L	100	1.29	0.34	0.18	0.10	0.08	0.09	0.10	0.13	0.17	0.24	0.35	0.55	0.83
O	95	0.15	0.06	0.05	0.07	0.09	0.11	0.12	0.15	0.18	0.25	0.37	0.58	0.87
N	90	0.16	0.05	0.04	0.05	0.06	0.08	0.11	0.16	0.23	0.36	0.55	0.82	1.19
G	85	0.31	0.07	0.04	0.03	0.05	0.07	0.12	0.17	0.28	0.47	0.76	1.23	1.75
	80	0.89	0.26	0.21	0.19	0.21	0.22	0.27	0.33	0.47	0.74	1.34	2.23	3.02
	75	X	X	X	0.08	2.25	2.11	2.52	3.29	4.75	8.50	977.4	X	X

TABLE 4b

CURRENT CONFIGURATION DOPPLER +- 1 HZ CHIRP +- 4 HZ/SEC

RMS TOTAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	55.57	36.75	5.12	6.09	7.66	10.93	12.30	17.09	20.34	24.51	257.1	397.2
	115	2.61	2.80	2.76	3.72	4.70	6.10	6.79	8.94	10.58	14.03	18.51	28.44	206.3
W	110	2.35	2.30	2.75	3.64	4.15	5.04	5.92	7.67	9.35	12.53	18.02	26.79	33.57
	105	8.29	3.11	2.93	3.23	3.63	4.82	5.81	6.85	8.89	11.83	14.00	24.09	29.08
L	100	2.00	2.14	2.12	2.67	3.28	4.12	4.97	6.12	7.76	9.78	11.90	14.65	24.90
O	95	1.58	1.58	1.73	2.13	2.62	3.56	4.86	6.35	7.88	9.91	12.94	16.18	25.86
N	90	1.09	1.21	1.39	1.78	2.33	3.36	5.04	6.32	8.69	11.40	14.02	24.40	30.06
G	85	1.14	1.43	1.76	2.32	2.79	3.68	5.45	6.61	8.36	11.66	14.42	25.84	32.64
	80	2.70	3.05	3.59	3.65	4.37	5.48	7.09	8.19	10.66	13.09	16.17	27.43	34.17
	75	X	X	X	3.35	8.49	9.74	11.24	16.21	18.73	22.60	125.7	179.4	206.8

TABLE 4c

CURRENT CONFIGURATION DOPPLER +- 1 HZ CHIRP +- 4 HZ/SEC

RMS IN-PLANE VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	45.75	32.54	0.19	0.18	0.18	0.33	0.36	0.45	0.54	0.68	203.1	313.0
	115	0.46	0.12	0.08	0.09	0.10	0.12	0.10	0.13	0.14	0.20	0.23	0.46	184.4
W	110	0.37	0.07	0.06	0.06	0.05	0.06	0.06	0.09	0.10	0.15	0.19	0.35	0.38
	105	0.23	0.09	0.05	0.05	0.05	0.06	0.06	0.07	0.11	0.13	0.14	0.28	0.31
L	100	0.17	0.11	0.05	0.04	0.04	0.05	0.05	0.06	0.09	0.10	0.11	0.14	0.25
O	95	0.13	0.06	0.05	0.04	0.04	0.04	0.05	0.08	0.09	0.11	0.16	0.22	0.26
N	90	0.06	0.05	0.04	0.04	0.04	0.06	0.07	0.08	0.14	0.16	0.19	0.27	0.30
G	85	0.11	0.06	0.05	0.06	0.06	0.07	0.09	0.10	0.12	0.20	0.22	0.31	0.35
	80	0.42	0.39	0.43	0.28	0.15	0.12	0.13	0.14	0.21	0.24	0.27	0.38	0.42
	75	X	X	X	2.51	0.82	0.69	0.63	1.82	2.18	2.71	101.5	132.6	170.2

TABLE 4d

CURRENT CONFIGURATION　　DOPPLER +- 1 HZ　　CHIRP +- 4 HZ/SEC

RMS PLANE-NORM.., VELOCITY ERRORS IN KM/MIN

SATELL.. .. ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	18.35	15.55	5.06	6.05	7.63	10.90	12.25	17.03	20.19	24.19	81.92	150.6
	115	2.44	2.78	2.74	3.69	4.67	6.08	6.77	8.89	10.55	13.97	18.49	28.26	90.96
W	110	2.23	2.28	2.73	3.62	4.13	5.03	5.90	7.65	9.33	12.50	17.99	26.69	33.45
	105	8.15	3.09	2.91	3.22	3.62	4.80	5.79	6.83	8.86	11.81	13.98	24.04	29.01
L	100	1.92	2.10	2.10	2.65	3.26	4.10	4.96	6.10	7.74	9.76	11.88	14.62	24.86
O	95	1.53	1.55	1.71	2.11	2.60	3.55	4.85	6.34	7.86	9.89	12.91	16.15	25.82
N	90	1.06	1.19	1.37	1.76	2.31	3.35	5.02	6.30	8.67	11.38	13.99	24.36	30.02
G	85	1.13	1.42	1.75	2.30	2.78	3.67	5.43	6.59	8.34	11.64	14.40	25.79	32.57
	80	2.56	2.93	3.46	3.57	4.33	5.45	7.05	8.17	10.63	13.07	16.15	27.34	34.05
	75	X	X	X	1.91	8.43	9.68	11.19	15.72	17.99	21.39	63.55	91.49	64.50

TABLE 5a

CURRENT + 3 35DB D-O OOPS   DOPPLER +- 20 HZ   NO CHIRP

RMS TOTAL POSITION ERRORS IN KM

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	55.56	44.06	1.46	0.65	0.46	0.50	0.60	0.84	1.35	2.81	X	X
	115	5.97	0.48	0.19	0.09	0.09	0.13	0.19	0.29	0.46	0.85	1.58	4.73	X
W	110	3.23	0.35	0.15	0.09	0.09	0.11	0.14	0.19	0.28	0.48	1.14	1.98	2.68
	105	10.61	3.45	0.46	0.13	0.09	0.09	0.11	0.14	0.21	0.32	0.51	0.98	1.32
L	100	1.27	0.34	0.18	0.10	0.08	0.09	0.10	0.13	0.17	0.24	0.35	0.55	0.83
O	95	0.15	0.06	0.05	0.07	0.09	0.11	0.12	0.15	0.18	0.25	0.37	0.58	0.87
N	90	0.16	0.05	0.04	0.05	0.06	0.08	0.11	0.16	0.23	0.36	0.55	0.82	1.18
G	85	0.31	0.07	0.04	0.03	0.04	0.07	0.12	0.17	0.28	0.47	0.76	1.23	1.75
	80	0.90	0.26	0.21	0.19	0.21	0.22	0.27	0.33	0.47	0.74	1.34	2.23	3.02
	75	X	X	X	207.5	2.27	2.12	2.53	3.29	4.74	8.50	995.3	X	X

TABLE 5b

CURRENT + 3 35DB D-O OOPS DOPPLER +- 20 HZ NO CHIRP

RMS TOTAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	64.84	49.46	8.43	5.18	5.40	6.19	9.84	11.10	12.45	13.91	101.1	169.8
	115	22.09	7.52	5.03	3.48	3.69	4.04	3.86	4.21	6.14	7.13	8.74	11.92	169.4
W	110	6.26	3.62	2.60	2.75	2.54	2.69	2.98	3.52	4.71	6.14	7.87	10.62	11.85
	105	4.16	2.14	1.94	1.95	2.07	2.49	2.83	3.19	4.71	5.60	6.46	9.69	10.90
L	100	4.47	2.62	1.73	1.53	1.64	1.86	2.49	2.99	4.41	5.25	6.14	7.05	10.21
O	95	2.73	1.68	1.60	1.69	1.78	2.08	2.46	3.73	4.54	5.39	7.40	8.77	10.39
N	90	2.05	1.60	1.60	1.73	1.97	2.39	2.92	4.21	5.70	6.84	8.00	9.76	10.96
G	85	7.56	4.92	2.47	2.50	2.48	2.81	3.35	3.74	5.82	7.59	8.79	10.69	11.92
	80	13.41	10.10	10.54	5.66	4.47	4.30	5.52	6.18	7.63	8.72	10.20	11.98	13.19
	75	X	X	X	X	14.36	11.62	11.83	14.17	15.25	16.86	76.27	88.63	79.50

TABLE 5c

CURRENT + 3 35DB D-O OOPS DOPPLER +- 20 HZ NO CHIRP

RMS IN-PLANE VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	25.18	16.39	3.19	2.50	2.49	3.47	4.50	5.59	6.24	6.90	19.75	28.80
	115	2.51	2.17	1.51	1.36	1.45	1.61	1.57	2.04	2.59	3.64	4.21	7.55	21.97
W	110	2.94	1.11	0.82	0.88	0.86	0.93	1.04	1.49	1.74	2.90	3.50	6.06	6.57
	105	2.63	1.04	0.90	0.84	0.85	0.95	1.05	1.15	2.01	2.43	2.74	4.89	5.39
L	100	2.50	1.45	0.78	0.65	0.65	0.69	0.79	0.94	1.62	1.85	2.10	2.58	4.40
O	95	1.52	0.81	0.69	0.64	0.55	0.61	0.80	1.26	1.73	1.99	2.94	4.01	4.70
N	90	1.08	0.73	0.61	0.61	0.67	0.92	1.17	1.40	2.58	3.01	3.46	4.86	5.41
G	85	1.51	1.04	0.92	0.95	0.96	1.10	1.50	1.68	2.16	3.56	4.05	5.67	6.24
	80	7.49	4.23	4.18	2.68	1.95	1.87	2.22	2.49	3.83	4.32	4.87	6.70	7.30
	75	X	X	X	X	7.86	5.41	5.22	7.03	7.45	8.14	19.07	21.98	20.59

TABLE 5d

CURRENT + 3 35DB D-O OOPS DOPPLER +- 20 HZ NO CHIRP

RMS PLANE-NORMAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	59.53	33.88	6.42	3.60	3.86	4.21	6.53	7.20	8.11	9.10	86.88	145.0
	115	21.85	6.43	4.27	2.75	2.91	3.18	3.20	3.40	5.00	5.78	6.94	8.21	151.1
W	110	4.06	2.84	1.97	2.14	2.08	2.27	2.55	2.92	4.13	5.15	6.48	7.78	8.79
	105	1.98	1.31	1.33	1.47	1.64	1.99	2.30	2.63	3.98	4.84	5.66	7.46	8.45
L	100	2.57	1.36	1.17	1.13	1.28	1.52	2.07	2.49	3.77	4.55	5.38	6.30	8.24
O	95	1.60	1.11	1.14	1.30	1.45	1.74	2.09	3.21	3.84	4.63	5.85	6.79	8.32
N	90	1.38	1.10	1.20	1.39	1.62	1.95	2.37	3.57	4.36	5.26	6.17	7.65	8.64
G	85	7.07	4.49	2.04	2.05	2.09	2.35	2.72	3.06	4.70	5.61	6.53	8.05	9.05
	80	10.69	7.54	7.93	3.53	3.17	3.22	4.34	4.80	5.40	6.20	7.31	8.59	9.56
	75	X	X	X	120.3	7.43	7.11	7.53	8.36	9.11	10.18	61.02	70.45	63.70

TABLE 6a

CURRENT + 3 35DB D-O OOPS DOPPLER +- 10 HZ NO CHIRP

RMS TOTAL POSITION ERRORS IN KM

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	55.56	43.61	1.46	0.65	0.46	0.50	0.60	0.84	1.35	2.81	923.0	X
	115	5.97	0.48	0.18	0.09	0.09	0.13	0.19	0.29	0.46	0.85	1.58	4.73	X
W	110	3.23	0.35	0.15	0.09	0.09	0.11	0.14	0.19	0.28	0.48	1.14	1.98	2.68
	105	10.58	3.15	0.46	0.13	0.09	0.09	0.11	0.14	0.21	0.32	0.51	0.98	1.32
L	100	1.22	0.34	0.18	0.10	0.08	0.09	0.10	0.13	0.17	0.24	0.35	0.55	0.83
O	95	0.15	0.06	0.05	0.07	0.09	0.11	0.12	0.15	0.18	0.25	0.37	0.58	0.87
N	90	0.16	0.05	0.04	0.05	0.06	0.08	0.11	0.16	0.23	0.36	0.55	0.82	1.18
G	85	0.31	0.07	0.04	0.03	0.04	0.07	0.12	0.17	0.28	0.47	0.76	1.23	1.75
	80	0.90	0.26	0.21	0.19	0.21	0.22	0.26	0.33	0.47	0.74	1.34	2.23	3.02
	75	X	X	X	207.5	2.27	2.12	2.53	3.28	4.74	8.50	560.9	768.5	X

TABLE 6b

CURRENT + 3 35DB D-O OOPS DOPPLER +- 10 HZ NO CHIRP

RMS TOTAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	64.83	49.47	4.54	2.64	2.73	3.13	5.06	5.69	6.38	7.10	63.26	87.59
	115	21.98	3.95	2.80	1.80	1.89	2.07	1.96	2.14	3.14	3.63	4.44	6.03	85.01
W	110	3.42	1.83	1.31	1.39	1.29	1.37	1.51	1.79	2.40	3.12	4.00	5.37	5.97
	105	2.32	1.07	0.97	0.98	1.05	1.27	1.44	1.62	2.41	2.85	3.28	4.92	5.52
L	100	2.27	1.32	0.87	0.77	0.83	0.95	1.27	1.53	2.29	2.72	3.16	3.60	5.22
O	95	1.49	0.89	0.86	0.90	0.93	1.07	1.26	1.94	2.36	2.78	3.81	4.48	5.29
N	90	1.29	0.94	0.96	0.98	1.08	1.27	1.52	2.23	2.98	3.52	4.08	4.95	5.54
G	85	3.98	2.96	1.31	1.36	1.34	1.50	1.75	1.94	3.07	3.93	4.49	5.42	6.02
	80	11.86	8.28	8.94	3.12	2.35	2.22	2.89	3.23	3.96	4.48	5.21	6.08	6.66
	75	X	X	X	X	7.36	5.99	6.12	7.34	7.92	8.74	42.65	53.39	64.73

TABLE 6c

CURRENT + 3 35DB D-O OOPS DOPPLER +- 10 HZ NO CHIRP

RMS IN-PLANE VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	25.17	16.29	1.64	1.26	1.25	1.74	2.28	2.83	3.15	3.48	11.40	14.78
	115	1.27	1.09	0.77	0.69	0.73	0.81	0.79	1.02	1.30	1.82	2.11	3.79	11.02
W	110	1.51	0.56	0.41	0.44	0.43	0.47	0.52	0.75	0.87	1.45	1.75	3.04	3.29
	105	1.33	0.52	0.45	0.42	0.43	0.48	0.52	0.58	1.01	1.22	1.37	2.45	2.70
L	100	1.27	0.73	0.39	0.33	0.33	0.34	0.40	0.47	0.81	0.93	1.05	1.29	2.21
O	95	0.80	0.42	0.35	0.33	0.28	0.31	0.40	0.63	0.87	1.00	1.48	2.01	2.36
N	90	0.56	0.38	0.33	0.31	0.34	0.46	0.59	0.71	1.31	1.52	1.74	2.43	2.71
G	85	0.76	0.53	0.46	0.48	0.48	0.55	0.76	0.85	1.10	1.80	2.04	2.84	3.13
	80	4.01	2.58	2.63	1.42	1.00	0.94	1.12	1.26	1.94	2.18	2.45	3.36	3.65
	75	X	X	X	X	4.00	2.76	2.67	3.59	3.82	4.16	10.66	13.21	16.12

TABLE 6d

CURRENT + 3 35DB D-O OOPS DOPPLER +- 10 HZ NO CHIRP

RMS PLANE-NORMAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	59.53	33.48	3.48	1.84	1.96	2.13	3.36	3.71	4.17	4.66	54.19	74.77
	115	21.85	3.39	2.39	1.43	1.50	1.63	1.63	1.73	2.57	2.95	3.54	4.17	75.80
W	110	2.28	1.44	1.00	1.09	1.05	1.16	1.30	1.49	2.11	2.63	3.30	3.95	4.44
	105	1.31	0.66	0.67	0.74	0.83	1.01	1.17	1.34	2.04	2.47	2.88	3.80	4.29
L	100	1.35	0.69	0.59	0.57	0.65	0.78	1.06	1.28	1.97	2.36	2.77	3.23	4.22
O	95	0.89	0.59	0.61	0.69	0.76	0.90	1.07	1.68	2.00	2.40	3.03	3.48	4.26
N	90	0.95	0.67	0.73	0.80	0.90	1.04	1.24	1.90	2.29	2.72	3.15	3.90	4.38
G	85	3.73	2.72	1.09	1.12	1.14	1.26	1.43	1.60	2.49	2.92	3.34	4.10	4.58
	80	10.69	6.39	6.92	1.96	1.68	1.67	2.28	2.52	2.82	3.20	3.75	4.38	4.84
	75	X	X	X	120.3	3.82	3.68	3.91	4.35	4.75	5.31	33.74	41.80	50.15

TABLE 7a

CURRENT + 3 35DB D-O OOPS   DOPPLER +- 10 HZ   CHIRP +- 10 HZ/SEC

RMS TOTAL POSITION ERRORS IN KM

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	38.84	35.27	1.46	0.65	0.46	0.50	0.60	0.84	1.35	2.81	839.9	X
	115	5.32	0.48	0.18	0.09	0.09	0.13	0.19	0.29	0.46	0.85	1.58	4.73	X
W	110	2.94	0.35	0.15	0.09	0.09	0.11	0.14	0.19	0.28	0.48	1.14	1.98	2.68
	105	6.47	3.11	0.46	0.13	0.09	0.09	0.11	0.14	0.21	0.32	0.51	0.9[illegible]	1.32
L	100	1.11	0.34	0.18	0.10	0.08	0.09	0.10	0.13	0.17	0.24	0.35	0.55	0.83
O	95	0.15	0.06	0.05	0.07	0.09	0.11	0.12	0.15	0.18	0.25	0.37	0.58	0.87
N	90	0.16	0.05	0.04	0.05	0.06	0.08	0.11	0.16	0.23	0.36	0.55	0.82	1.18
G	85	0.31	0.07	0.04	0.03	0.04	0.07	0.12	0.17	0.28	0.47	0.76	1.23	1.75
	80	0.89	0.26	0.21	0.19	0.21	0.22	0.26	0.33	0.47	0.74	1.34	2.23	3.02
	75	X	X	X	0.01	2.27	2.12	2.53	3.28	4.74	8.50	541.9	705.0	945.5

TABLE 7b

CURRENT + 3 35DB D-O OOPS   DOPPLER +- 10 HZ   CHIRP +- 10 HZ/SEC

RMS TOTAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	56.72	29.40	4.40	2.62	2.72	3.12	5.03	5.68	6.36	7.08	57.50	73.90
	115	5.96	3.53	2.65	1.78	1.88	2.05	1.96	2.13	3.13	3.62	4.43	6.01	68.67
W	110	3.02	1.77	1.29	1.38	1.28	1.36	1.51	1.78	2.39	3.11	3.99	5.36	5.96
	105	2.07	1.07	0.97	0.98	1.04	1.26	1.43	1.62	2.40	2.85	3.27	4.91	5.51
L	100	2.12	1.30	0.87	0.77	0.83	0.94	1.27	1.53	2.29	2.71	3.15	3.59	5.21
O	95	1.46	0.88	0.85	0.90	0.92	1.07	1.25	1.93	2.35	2.77	3.80	4.47	5.28
N	90	1.25	0.93	0.95	0.97	1.08	1.26	1.51	2.22	2.97	3.51	4.06	4.95	5.53
G	85	2.37	2.33	1.27	1.34	1.33	1.48	1.74	1.93	3.05	3.92	4.47	5.41	6.01
	80	6.89	6.48	7.20	3.06	2.33	2.21	2.87	3.21	3.95	4.47	5.19	6.07	6.65
	75	X	X	X	0.12	7.27	5.94	6.07	7.30	7.88	8.70	41.15	48.89	59.97

TABLE 7c

CURRENT + 3 35DB D-O OOPS DOPPLER +- 10 HZ CHIRP +- 10 HZ/SEC

RMS IN-PLANE VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	43.73	9.63	1.63	1.26	1.25	1.74	2.28	2.82	3.15	3.48	10.83	13.15
	115	1.27	1.08	0.76	0.68	0.73	0.81	0.79	1.02	1.30	1.82	2.11	3.79	9.66
W	110	1.44	0.55	0.41	0.44	0.43	0.47	0.52	0.75	0.87	1.45	1.75	3.04	3.29
	105	1.31	0.52	0.45	0.42	0.43	0.48	0.52	0.58	1.01	1.22	1.37	2.45	2.70
L	100	1.25	0.72	0.39	0.33	0.33	0.34	0.40	0.47	0.81	0.93	1.05	1.29	2.21
O	95	0.79	0.42	0.35	0.33	0.28	0.31	0.40	0.63	0.87	1.00	1.48	2.01	2.36
N	90	0.56	0.38	0.33	0.31	0.34	0.46	0.59	0.71	1.30	1.52	1.74	2.43	2.71
G	85	0.74	0.52	0.46	0.48	0.48	0.55	0.76	0.85	1.10	1.80	2.04	2.84	3.13
	80	3.99	2.36	2.40	1.41	0.99	0.94	1.12	1.26	1.94	2.18	2.45	3.36	3.65
	75	X	X	X	0.08	3.97	2.75	2.65	3.58	3.81	4.15	10.46	12.45	15.36

TABLE 7d

CURRENT + 3 35DB D-O OOPS DOPPLER +- 10 HZ CHIRP +- 10 HZ/SEC

RMS PLANE-NORMAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	26.49	20.26	3.36	1.82	1.94	2.12	3.34	3.69	4.15	4.65	49.17	62.98
	115	5.75	3.00	2.26	1.41	1.48	1.62	1.62	1.72	2.56	2.95	3.53	4.16	61.14
W	110	1.97	1.38	0.98	1.08	1.05	1.15	1.29	1.48	2.10	2.62	3.29	3.94	4.43
	105	1.00	0.66	0.67	0.74	0.83	1.01	1.17	1.33	2.03	2.47	2.87	3.79	4.28
L	100	1.19	0.68	0.59	0.57	0.65	0.77	1.06	1.27	1.96	2.36	2.76	3.22	4.21
O	95	0.86	0.59	0.61	0.69	0.75	0.90	1.07	1.67	1.99	2.39	3.01	3.47	4.25
N	90	0.91	0.66	0.72	0.79	0.89	1.03	1.24	1.89	2.28	2.71	3.14	3.89	4.37
G	85	2.15	2.12	1.05	1.10	1.12	1.25	1.43	1.59	2.48	2.91	3.33	4.09	4.57
	80	5.50	4.89	5.50	1.92	1.66	1.65	2.26	2.50	2.81	3.19	3.73	4.37	4.83
	75	X	X	X	0.07	3.76	3.64	3.87	4.32	4.72	5.28	32.48	38.18	46.24

TABLE 8a

CURRENT + 3 35DB D-O OOPS   DOPPLER +- 1 HZ   CHIRP +- 4 HZ/SEC

RMS TOTAL POSITION ERRORS IN KM

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	36.11	32.00	1.40	0.63	0.46	0.49	0.60	0.84	1.35	2.80	98.98	143.7
	115	4.61	0.48	0.18	0.09	0.09	0.13	0.19	0.29	0.46	0.84	1.58	4.73	127.9
W	110	2.08	0.35	0.15	0.09	0.09	0.11	0.14	0.19	0.28	0.48	1.14	1.98	2.68
	105	3.20	0.64	0.31	0.12	0.09	0.09	0.11	0.14	0.21	0.32	0.51	0.98	1.32
L	100	0.37	0.19	0.15	0.10	0.08	0.09	0.10	0.13	0.17	0.24	0.35	0.55	0.83
O	95	0.12	0.06	0.05	0.07	0.09	0.11	0.12	0.15	0.18	0.25	0.37	0.58	0.87
N	90 *	0.16	0.05	0.04	0.05	0.06	0.08	0.11	0.16	0.23	0.36	0.55	0.82	1.18
G	85	0.30	0.07	0.04	0.03	0.04	0.07	0.12	0.17	0.28	0.47	0.76	1.23	1.75
	80	0.89	0.26	0.21	0.18	0.21	0.22	0.26	0.33	0.47	0.74	1.34	2.23	3.02
	75	X	X	X	0.01	2.23	2.09	2.49	3.24	4.67	8.29	59.75	84.01	116.0

TABLE 8b

CURRENT + 3 35DB D-O OOPS DOPPLER +- 1 HZ CHIRP +- 4 HZ/SEC

RMS TOTAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	55.57	16.68	0.50	0.27	0.28	0.32	0.51	0.58	0.65	0.73	6.76	8.87
	115	2.60	0.40	0.29	0.18	0.19	0.21	0.20	0.22	0.32	0.37	0.46	0.70	8.51
W	110	1.02	0.22	0.14	0.14	0.13	0.14	0.15	0.18	0.24	0.32	0.41	0.56	0.63
	105	0.41	0.11	0.10	0.10	0.11	0.13	0.15	0.16	0.24	0.29	0.33	0.50	0.56
L	100	0.25	0.14	0.09	0.08	0.09	0.10	0.13	0.16	0.23	0.28	0.32	0.37	0.53
O	95	0.18	0.09	0.09	0.09	0.10	0.11	0.13	0.20	0.24	0.28	0.39	0.45	0.54
N	90*	0.18	0.10	0.10	0.10	0.11	0.13	0.16	0.23	0.30	0.36	0.41	0.50	0.56
G	85	0.41	0.31	0.13	0.14	0.14	0.15	0.18	0.20	0.31	0.40	0.45	0.55	0.61
	80	2.71	1.26	1.48	0.32	0.24	0.23	0.29	0.33	0.40	0.45	0.53	0.62	0.68
	75	X	X	X	0.19	0.75	0.62	0.64	0.76	0.83	0.96	4.53	5.81	7.35

TABLE 8c

CURRENT + 3 35DB D-O OOPS DOPPLER +- 1 HZ CHIRP +- 4 HZ/SEC

RMS IN-PLANE VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	45.75	4.61	0.17	0.13	0.13	0.18	0.23	0.28	0.32	0.35	1.20	1.49
	115	0.46	0.11	0.08	0.07	0.07	0.08	0.08	0.10	0.13	0.18	0.21	0.38	1.11
W	110	0.29	0.07	0.05	0.05	0.04	0.05	0.05	0.08	0.09	0.15	0.18	0.30	0.33
	105	0.15	0.06	0.05	0.04	0.04	0.05	0.05	0.06	0.10	0.12	0.14	0.25	0.27
L	100	0.15	0.08	0.04	0.03	0.03	0.04	0.04	0.05	0.08	0.09	0.11	0.13	0.22
O	95	0.09	0.04	0.04	0.03	0.03	0.03	0.04	0.06	0.09	0.10	0.15	0.20	0.24
N	90	0.06	0.04	0.03	0.03	0.03	0.05	0.06	0.07	0.13	0.15	0.17	0.24	0.27
G	85	0.11	0.06	0.05	0.05	0.05	0.06	0.08	0.09	0.11	0.18	0.20	0.28	0.31
	80	0.42	0.33	0.36	0.15	0.10	0.10	0.11	0.13	0.19	0.22	0.25	0.34	0.37
	75	X	X	X	0.14	0.41	0.28	0.27	0.37	0.40	0.45	1.13	1.44	1.83

TABLE 8d

CURRENT + 3 35DB D-O OOPS   DOPPLER +- 1 HZ   CHIRP +- 4 HZ/SEC

RMS PLANE-NORMAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	18.35	12.31	0.40	0.19	0.20	0.22	0.34	0.38	0.43	0.50	5.79	7.57
	115	2.44	0.35	0.25	0.14	0.15	0.16	0.17	0.18	0.26	0.31	0.38	0.52	7.59
W	110	0.76	0.17	0.10	0.11	0.11	0.12	0.13	0.15	0.22	0.27	0.35	0.42	0.48
	105	0.31	0.07	0.08	0.08	0.09	0.10	0.12	0.14	0.21	0.25	0.29	0.39	0.44
L	100	0.17	0.07	0.06	0.06	0.07	0.08	0.11	0.13	0.20	0.24	0.28	0.33	0.43
O	95	0.11	0.06	0.06	0.07	0.08	0.09	0.11	0.17	0.20	0.24	0.31	0.35	0.43
N	90	0.15	0.07	0.08	0.09	0.09	0.11	0.13	0.20	0.23	0.28	0.32	0.40	0.45
G	85	0.37	0.29	0.11	0.12	0.12	0.13	0.15	0.16	0.26	0.30	0.34	0.42	0.47
	80	2.56	0.98	1.16	0.20	0.17	0.17	0.23	0.26	0.29	0.33	0.39	0.46	0.51
	75	X	X	X	0.11	0.43	0.41	0.44	0.50	0.58	0.75	3.56	4.52	5.66

TABLE 9a

CURRENT + 3 35DB OOPS DOPPLER +- 20 HZ NO CHIRP

RMS TOTAL POSITION ERRORS IN KM

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	55.56	0.25	0.09	0.05	0.05	0.06	0.06	0.07	0.09	0.11	0.15	0.21
	115	5.97	0.07	0.03	0.02	0.02	0.03	0.03	0.03	0.04	0.05	0.07	0.10	0.15
W	110	0.06	0.04	0.01	0.01	0.01	0.01	0.02	0.02	0.03	0.04	0.05	0.08	0.11
	105	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.04	0.05	0.07
L	100	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.04	0.05
O	95	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.04	0.06
N	90	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.04	0.06	0.09
G	85	0.07	0.02	0.01	0.01	0.01	0.02	0.02	0.02	0.03	0.04	0.06	0.08	0.12
	80	0.90	0.11	0.06	0.02	0.02	0.03	0.03	0.04	0.05	0.06	0.08	0.11	0.16
	75	X	X	X	207.5	0.08	0.06	0.07	0.07	0.08	0.10	0.13	0.17	0.23

TABLE 9b

CURRENT + 3 35DB OOPS DOPPLER +- 20 HZ NO CHIRP

RMS TOTAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	64.84	9.46	3.47	2.58	2.44	2.95	2.96	3.12	3.27	3.65	4.98	6.73
	115	22.09	4.06	1.90	1.52	1.51	1.49	1.46	1.66	1.83	2.22	2.65	3.64	5.13
W	110	4.66	1.36	1.09	1.02	0.93	0.91	0.96	1.13	1.35	1.79	2.24	3.11	4.20
	105	1.84	0.95	0.76	0.65	0.63	0.69	0.80	0.94	1.22	1.47	1.79	2.32	2.97
L	100	0.64	0.66	0.61	0.53	0.50	0.52	0.60	0.75	0.92	1.09	1.32	1.63	2.12
O	95	0.90	0.66	0.58	0.46	0.43	0.47	0.60	0.83	1.02	1.21	1.48	1.86	2.38
N	90	0.99	0.69	0.65	0.53	0.50	0.56	0.71	0.89	1.23	1.54	1.92	2.56	3.34
G	85	3.44	1.25	1.03	0.90	0.82	0.81	0.91	1.08	1.37	1.76	2.21	3.00	4.04
	80	13.41	4.40	3.97	1.94	1.58	1.52	1.69	1.80	2.05	2.28	2.72	3.56	4.66
	75	X	X	X	X	6.20	4.05	3.50	3.65	3.53	3.64	4.32	5.13	6.89

TABLE 9c

CURRENT + 3 35DB OOPS   DOPPLER +- 20 HZ   NO CHIRP

RMS IN-PLANE VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	25.18	6.14	2.74	2.13	2.02	2.50	2.51	2.64	2.73	3.04	4.16	5.61
	115	2.51	2.02	1.37	1.19	1.19	1.18	1.13	1.30	1.45	1.77	2.09	2.89	4.14
W	110	2.60	0.93	0.76	0.72	0.66	0.65	0.69	0.83	1.00	1.37	1.69	2.37	3.14
	105	1.24	0.62	0.49	0.43	0.44	0.50	0.57	0.65	0.90	1.08	1.28	1.69	2.14
L	100	0.62	0.41	0.45	0.40	0.38	0.38	0.42	0.51	0.66	0.76	0.90	1.09	1.45
O	95	0.62	0.44	0.42	0.36	0.34	0.35	0.43	0.60	0.73	0.86	1.05	1.33	1.70
N	90	0.73	0.51	0.48	0.40	0.37	0.41	0.52	0.65	0.92	1.15	1.43	1.91	2.46
G	85	1.47	0.96	0.81	0.71	0.64	0.63	0.71	0.83	1.05	1.37	1.71	2.32	3.09
	80	7.49	3.79	3.54	1.73	1.36	1.27	1.38	1.45	1.67	1.84	2.18	2.85	3.70
	75	X	X	X	X	4.92	3.34	2.98	3.20	3.05	3.10	3.66	4.33	5.67

TABLE 9d

CURRENT + 3 35DB OOPS   DOPPLER +- 20 HZ   NO CHIRP

RMS PLANE-NORMAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	59.53	5.11	2.03	1.32	1.22	1.24	1.28	1.32	1.47	1.75	2.28	3.25
	115	21.85	3.27	1.20	0.81	0.75	0.72	0.73	0.80	0.92	1.12	1.42	1.88	2.73
W	110	2.92	0.72	0.56	0.51	0.49	0.50	0.54	0.62	0.79	1.02	1.33	1.79	2.50
	105	0.99	0.55	0.48	0.40	0.38	0.42	0.50	0.60	0.75	0.93	1.16	1.46	1.92
L	100	0.11	0.29	0.32	0.30	0.30	0.33	0.41	0.52	0.63	0.76	0.94	1.17	1.48
O	95	0.37	0.22	0.18	0.21	0.23	0.30	0.40	0.55	0.67	0.82	1.01	1.25	1.60
N	90	0.38	0.30	0.27	0.26	0.28	0.34	0.42	0.56	0.74	0.95	1.22	1.60	2.14
G	85	2.96	0.65	0.51	0.43	0.40	0.42	0.48	0.59	0.78	0.99	1.29	1.74	2.46
	80	10.69	2.14	1.59	0.78	0.69	0.73	0.84	0.91	1.00	1.16	1.45	1.90	2.65
	75	X	X	X	120.3	2.64	1.92	1.70	1.60	1.60	1.73	2.05	2.59	3.71

TABLE 10a

CURRENT + 3 35DB OOPS   DOPPLER +- 10 HZ   NO CHIRP

RMS TOTAL POSITION ERRORS IN KM

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	55.56	0.25	0.09	0.05	0.05	0.06	0.06	0.07	0.09	0.11	0.15	0.21
	115	5.97	0.07	0.03	0.02	0.02	0.03	0.03	0.03	0.04	0.05	0.07	0.10	0.15
W	110	0.06	0.04	0.01	0.01	0.01	0.01	0.02	0.02	0.03	0.04	0.05	0.08	0.11
	105	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.04	0.05	0.07
L	100	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.04	0.05
O	95	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.04	0.06
N	90	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.04	0.06	0.09
G	85	0.07	0.02	0.01	0.01	0.01	0.02	0.02	0.02	0.03	0.04	0.06	0.08	0.12
	80	0.90	0.11	0.06	0.02	0.02	0.03	0.03	0.04	0.05	0.06	0.08	0.11	0.16
	75	X	X	X	207.5	0.08	0.06	0.07	0.07	0.08	0.10	0.13	0.17	0.23

TABLE 10b

CURRENT + 3 35DB OOPS DOPPLER +- 10 HZ NO CHIRP

RMS TOTAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	64.83	8.54	2.57	1.75	1.65	1.92	2.10	2.32	2.52	2.87	4.31	5.56
	115	21.98	2.99	1.38	1.02	0.99	1.00	1.01	1.22	1.38	1.78	2.10	3.15	4.40
W	110	2.80	0.86	0.70	0.66	0.64	0.67	0.72	0.90	1.08	1.49	1.84	2.71	3.49
	105	1.15	0.70	0.59	0.51	0.52	0.58	0.67	0.78	1.05	1.27	1.53	2.09	2.62
L	100	0.53	0.44	0.44	0.41	0.40	0.43	0.52	0.65	0.82	0.98	1.18	1.45	1.95
O	95	0.62	0.39	0.33	0.30	0.32	0.39	0.51	0.71	0.88	1.06	1.34	1.70	2.15
N	90	0.63	0.43	0.38	0.35	0.38	0.47	0.60	0.75	1.06	1.33	1.66	2.23	2.83
G	85	2.67	0.82	0.67	0.60	0.58	0.61	0.74	0.87	1.10	1.49	1.85	2.54	3.31
	80	11.86	3.20	2.60	1.36	1.08	1.07	1.23	1.33	1.63	1.84	2.19	2.94	3.72
	75	X	X	X	X	4.63	3.10	2.74	2.87	2.84	3.00	3.64	4.40	5.88

TABLE 10c

CURRENT + 3 35DB OOPS DOPPLER +- 10 HZ NO CHIRP

RMS IN-PLANE VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	25.17	4.11	1.45	1.14	1.11	1.51	1.62	1.91	2.05	2.28	3.66	4.65
	115	1.27	1.05	0.70	0.63	0.65	0.69	0.69	0.87	1.02	1.35	1.57	2.41	3.53
W	110	1.40	0.48	0.40	0.41	0.40	0.41	0.45	0.60	0.71	1.06	1.29	1.96	2.41
	105	0.81	0.43	0.35	0.31	0.32	0.36	0.40	0.46	0.71	0.85	0.99	1.44	1.75
L	100	0.50	0.28	0.28	0.26	0.25	0.26	0.29	0.36	0.52	0.61	0.71	0.87	1.24
O	95	0.46	0.26	0.24	0.21	0.21	0.24	0.31	0.45	0.58	0.68	0.87	1.14	1.43
N	90	0.45	0.29	0.25	0.23	0.25	0.32	0.41	0.50	0.77	0.94	1.13	1.57	1.92
G	85	0.74	0.49	0.43	0.42	0.41	0.44	0.55	0.63	0.78	1.12	1.35	1.88	2.33
	80	4.01	2.03	1.90	1.00	0.80	0.78	0.91	0.98	1.30	1.44	1.67	2.27	2.75
	75	X	X	X	X	2.98	1.98	1.81	2.20	2.20	2.31	2.95	3.40	4.17

TABLE 10d

CURRENT + 3 35DB OOPS DOPPLER +- 10 HZ NO CHIRP

RMS PLANE-NORMAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	59.53	5.09	1.81	1.11	1.06	1.09	1.19	1.23	1.38	1.63	2.11	2.88
	115	21.85	2.52	1.10	0.72	0.67	0.65	0.67	0.74	0.87	1.06	1.33	1.75	2.48
W	110	1.78	0.62	0.48	0.45	0.44	0.46	0.50	0.58	0.74	0.96	1.25	1.66	2.25
	105	0.57	0.40	0.37	0.34	0.35	0.39	0.46	0.56	0.72	0.89	1.10	1.39	1.79
L	100	0.11	0.21	0.27	0.27	0.27	0.31	0.39	0.49	0.61	0.73	0.90	1.12	1.42
O	95	0.26	0.14	0.11	0.19	0.22	0.29	0.38	0.53	0.64	0.78	0.96	1.19	1.52
N	90	0.27	0.23	0.23	0.23	0.26	0.32	0.40	0.53	0.70	0.90	1.15	1.50	1.96
G	85	2.45	0.63	0.46	0.39	0.38	0.40	0.46	0.56	0.74	0.94	1.21	1.62	2.22
	80	10.69	2.09	1.56	0.72	0.64	0.68	0.79	0.86	0.94	1.10	1.36	1.77	2.38
	75	X	X	X	120.3	2.25	1.74	1.57	1.48	1.50	1.62	1.91	2.39	3.30

TABLE 11a

CURRENT + 3 35DB OOPS DOPPLER +- 10 HZ CHIRP +- 10 HZ/SEC

RMS TOTAL POSITION ERRORS IN KM

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	38.84	0.25	0.09	0.05	0.05	0.06	0.06	0.07	0.09	0.11	0.15	0.21
	115	5.32	0.07	0.03	0.02	0.02	0.03	0.03	0.03	0.04	0.05	0.07	0.10	0.15
W	110	0.06	0.04	0.01	0.01	0.01	0.01	0.02	0.02	0.03	0.04	0.05	0.08	0.11
	105	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.04	0.05	0.07
L	100	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.04	0.05
O	95	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.04	0.06
N	90	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.04	0.06	0.09
G	85	0.06	0.02	0.01	0.01	0.01	0.02	0.02	0.02	0.03	0.04	0.06	0.08	0.12
	80	0.89	0.11	0.06	0.02	0.02	0.03	0.03	0.04	0.05	0.06	0.08	0.11	0.16
	75	X	X	X	0.01	0.08	0.06	0.07	0.07	0.08	0.10	0.13	0.17	0.23

TABLE 11b

CURRENT + 3 35DB OOPS DOPPLER +- 10 HZ CHIRP +- 10 HZ/SEC

RMS TOTAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	56.72	8.15	2.56	1.75	1.65	1.92	2.10	2.32	2.52	2.87	4.31	5.56
	115	5.96	2.81	1.36	1.02	0.99	1.00	1.01	1.22	1.38	1.78	2.10	3.15	4.40
W	110	2.65	0.86	0.70	0.66	0.64	0.67	0.72	0.90	1.08	1.49	1.84	2.71	3.48
	105	1.15	0.70	0.59	0.51	0.51	0.58	0.67	0.78	1.05	1.27	1.53	2.09	2.62
L	100	0.53	0.44	0.44	0.40	0.40	0.43	0.52	0.65	0.82	0.98	1.18	1.45	1.95
O	95	0.62	0.39	0.33	0.30	0.32	0.39	0.51	0.71	0.88	1.06	1.34	1.70	2.15
N	90	0.62	0.43	0.38	0.35	0.38	0.47	0.60	0.75	1.06	1.33	1.66	2.23	2.83
G	85	2.02	0.82	0.67	0.60	0.58	0.61	0.74	0.87	1.10	1.49	1.85	2.54	3.31
	80	6.89	3.14	2.59	1.36	1.08	1.07	1.23	1.33	1.63	1.84	2.19	2.94	3.72
	75	X	X	X	0.12	4.61	3.10	2.74	2.87	2.84	3.00	3.64	4.40	5.88

TABLE 11c

CURRENT + 3 35DB COPS DOPPLER +- 10 HZ CHIRP +- 10 HZ/SEC

RMS IN-PLANE VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	43.73	4.01	1.44	1.14	1.11	1.51	1.62	1.91	2.05	2.28	3.66	4.65
	115	1.27	1.04	0.70	0.63	0.65	0.69	0.69	0.87	1.02	1.35	1.57	2.41	3.53
W	110	1.38	0.48	0.40	0.41	0.40	0.41	0.45	0.60	0.71	1.06	1.29	1.96	2.41
	105	0.81	0.43	0.35	0.31	0.32	0.36	0.40	0.46	0.71	0.85	0.99	1.44	1.75
L	100	0.50	0.28	0.28	0.26	0.25	0.26	0.29	0.36	0.52	0.61	0.71	0.87	1.24
O	95	0.46	0.26	0.24	0.21	0.21	0.24	0.31	0.45	0.58	0.68	0.87	1.14	1.43
N	90	0.45	0.29	0.25	0.23	0.25	0.32	0.41	0.50	0.77	0.94	1.13	1.57	1.92
G	85	0.74	0.49	0.43	0.42	0.41	0.44	0.55	0.63	0.78	1.12	1.35	1.88	2.33
	80	3.99	2.02	1.90	1.00	0.80	0.78	0.91	0.98	1.30	1.44	1.67	2.27	2.75
	75	X	X	X	0.08	2.98	1.98	1.81	2.20	2.20	2.31	2.95	3.40	4.17

TABLE 11d

CURRENT + 3 35DB OOPS   DOPPLER +- 10 HZ   CHIRP +- 10 HZ/SEC

RMS PLANE-NORMAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	26.49	4.81	1.79	1.11	1.06	1.08	1.19	1.23	1.38	1.63	2.11	2.88
	115	5.75	2.35	1.08	0.71	0.67	0.65	0.67	0.74	0.87	1.06	1.33	1.75	2.47
W	110	1.67	0.61	0.47	0.45	0.44	0.46	0.50	0.58	0.74	0.96	1.25	1.66	2.25
	105	0.57	0.40	0.37	0.34	0.35	0.39	0.46	0.56	0.72	0.89	1.10	1.39	1.79
L	100	0.11	0.21	0.27	0.27	0.27	0.31	0.39	0.49	0.61	0.73	0.90	1.12	1.42
O	95	0.26	0.14	0.11	0.19	0.22	0.29	0.38	0.53	0.64	0.78	0.96	1.19	1.52
N	90	0.27	0.23	0.23	0.23	0.26	0.32	0.40	0.53	0.70	0.90	1.15	1.50	1.96
G	85	1.79	0.62	0.46	0.39	0.38	0.40	0.45	0.56	0.74	0.94	1.21	1.62	2.22
	80	5.50	2.02	1.54	0.72	0.64	0.68	0.79	0.86	0.94	1.10	1.36	1.77	2.38
	75	X	X	X	0.07	2.24	1.73	1.57	1.48	1.50	1.62	1.91	2.39	3.30

TABLE 12a

CURRENT + 3 35DB OOPS   DOPPLER +- 1 HZ   CHIRP +- 4 HZ/SEC

RMS TOTAL POSITION ERRORS IN KM

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	36.11	0.25	0.09	0.05	0.05	0.06	0.06	0.07	0.09	0.11	0.15	0.21
	115	4.61	0.07	0.03	0.02	0.02	0.03	0.03	0.03	0.04	0.05	0.07	0.10	0.15
W	110	0.06	0.03	0.01	0.01	0.01	0.01	0.02	0.02	0.03	0.04	0.05	0.08	0.11
	105	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.04	0.05	0.07
L	100	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.04	0.05
O	95	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.04	0.06
N	90	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.04	0.06	0.09
G	85	0.06	0.02	0.01	0.01	0.01	0.02	0.02	0.02	0.03	0.04	0.06	0.08	0.12
	80	0.89	0.11	0.06	0.02	0.02	0.03	0.03	0.04	0.05	0.06	0.08	0.11	0.16
	75	X	X	X	0.01	0.08	0.06	0.07	0.07	0.08	0.10	0.13	0.17	0.23

TABLE 12b

CURRENT + 3 35DB OOPS DOPPLER +- 1 HZ CHIRP +- 4 HZ/SEC

RMS TOTAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	55.57	6.24	0.46	0.26	0.27	0.31	0.49	0.55	0.62	0.69	0.98	1.06
	115	2.60	0.40	0.28	0.18	0.19	0.20	0.19	0.21	0.31	0.35	0.43	0.59	0.92
W	110	0.32	0.18	0.13	0.14	0.13	0.13	0.15	0.18	0.23	0.31	0.39	0.53	0.59
	105	0.21	0.11	0.10	0.10	0.10	0.12	0.14	0.16	0.24	0.28	0.32	0.48	0.54
L	100	0.16	0.10	0.09	0.08	0.08	0.09	0.12	0.15	0.22	0.26	0.31	0.35	0.51
O	95	0.14	0.07	0.07	0.09	0.09	0.10	0.12	0.19	0.23	0.27	0.37	0.44	0.52
N	90	0.14	0.10	0.10	0.10	0.11	0.12	0.15	0.22	0.29	0.34	0.40	0.49	0.55
G	85	0.39	0.29	0.13	0.14	0.13	0.15	0.17	0.19	0.30	0.38	0.44	0.53	0.59
	80	2.71	1.15	1.20	0.31	0.23	0.22	0.28	0.32	0.39	0.44	0.51	0.60	0.66
	75	X	X	X	0.19	0.73	0.60	0.60	0.72	0.77	0.84	0.99	1.09	1.20

TABLE 12c

CURRENT + 3 35DB OOPS DOPPLER +- 1 HZ CHIRP +- 4 HZ/SEC

RMS IN-PLANE VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	45.75	2.39	0.17	0.13	0.13	0.17	0.23	0.28	0.31	0.34	0.71	0.75
	115	0.46	0.11	0.08	0.07	0.07	0.08	0.08	0.10	0.13	0.18	0.21	0.38	0.61
W	110	0.17	0.06	0.04	0.04	0.04	0.05	0.05	0.07	0.09	0.14	0.17	0.30	0.33
	105	0.14	0.05	0.04	0.04	0.04	0.05	0.05	0.06	0.10	0.12	0.14	0.24	0.27
L	100	0.11	0.06	0.04	0.03	0.03	0.03	0.04	0.05	0.08	0.09	0.10	0.13	0.22
O	95	0.08	0.04	0.03	0.03	0.03	0.03	0.04	0.06	0.09	0.10	0.15	0.20	0.23
N	90	0.06	0.04	0.03	0.03	0.03	0.05	0.06	0.07	0.13	0.15	0.17	0.24	0.27
G	85	0.09	0.05	0.05	0.05	0.05	0.06	0.08	0.08	0.11	0.18	0.20	0.28	0.31
	80	0.42	0.31	0.31	0.14	0.10	0.09	0.11	0.13	0.19	0.22	0.24	0.33	0.36
	75	X	X	X	0.14	0.40	0.27	0.26	0.35	0.37	0.41	0.54	0.58	0.63

TABLE 12d

CURRENT + 3 35DB OOPS DOPPLER +- 1 HZ CHIRP +- 4 HZ/SEC

RMS PLANE-NORMAL VELOCITY ERRORS IN KM/MIN

SATELLITE ALTITUDE IN KM

		100	200	300	500	700	1000	1300	1600	2000	2500	3000	3500	4000
	120	X	18.35	3.85	0.35	0.18	0.19	0.21	0.33	0.36	0.40	0.45	0.51	0.56
	115	2.44	0.34	0.24	0.14	0.15	0.16	0.16	0.17	0.25	0.29	0.35	0.41	0.53
W	110	0.20	0.14	0.10	0.11	0.10	0.11	0.13	0.15	0.21	0.26	0.32	0.39	0.44
	105	0.09	0.06	0.07	0.07	0.08	0.10	0.11	0.13	0.20	0.24	0.28	0.37	0.42
L	100	0.08	0.06	0.06	0.06	0.06	0.08	0.10	0.13	0.19	0.23	0.27	0.31	0.41
O	95	0.08	0.05	0.05	0.07	0.07	0.09	0.10	0.16	0.19	0.23	0.29	0.34	0.41
N	90	0.10	0.07	0.07	0.08	0.09	0.10	0.12	0.18	0.22	0.26	0.31	0.38	0.43
G	85	0.36	0.26	0.11	0.11	0.11	0.12	0.14	0.16	0.24	0.28	0.33	0.40	0.45
	80	2.56	0.90	0.94	0.20	0.17	0.16	0.22	0.25	0.27	0.31	0.37	0.43	0.48
	75	X	X	X	0.11	0.38	0.37	0.38	0.42	0.46	0.51	0.57	0.63	0.70

APPENDIX A

This appendix presents a brief discussion of how Doppler and Doppler rate (chirp) measurements can be used to improve the velocity and position determination in the NAVSPASUR system.

For a bi-static radar, the observed Doppler shift frequency ν_D is given by

$$\nu_D(t) = -\frac{\nu_0}{c}\left\{\vec{v}_s \cdot \left[\hat{r}_{ts} + \hat{r}_{rs}\left(1 + \frac{\vec{v}_s \cdot \hat{r}_{ts}}{c}\right)\right]\right\} \tag{A.1}$$

where ν_0 is the NAVSPASUR transmit frequency (216.98 MHz), c is the velocity of light, $\vec{v}_s$ is the velocity vector of the target satellite, and $\hat{r}_{ts}$, $\hat{r}_{rs}$ are the unit vectors to the satellite from the transmitter and receiver, respectively. Since for all orbits of interest $(\vec{v}_s \cdot \hat{r}_{ts}) << c$, equation (A.1) reduces to

$$\nu_D(t) \approx -\frac{\nu_0}{c}\left\{\vec{v}_s \cdot [\hat{r}_{ts} + \hat{r}_{rs}]\right\}. \tag{A.2}$$

Adopting an earth-centered cartesian coordinate system where $\hat{x}$, $\hat{y}$ are in the NAVSPASUR great circle plane and $\hat{z}$ is normal to the plane, equation (A.2) becomes

$$\nu_D(t_0) = -\frac{\nu_0}{c}\left\{\frac{v_x(x_s-x_t) + v_y(y_s-y_t) + v_z(z_s-z_t)}{|\vec{r}_{ts}|} + \frac{v_x(x_s-x_r) + v_y(y_s-y_r) + v_z(z_s-z_r)}{|\vec{r}_{rs}|}\right\} \tag{A.3}$$

where

$$|\vec{r}_{ts}| = \left[(x_s-x_t)^2 + (y_s-y_t)^2 + (z_s-z_t)^2\right]^{1/2}$$

$$|\vec{r}_{rs}| = \left[(x_s-x_r)^2 + (y_s-y_r)^2 + (z_s-z_r)^2\right]^{1/2}.$$

Here v_x, v_y, v_z are the components of the satellite velocity at time t_0 and x_s, y_s, z_s, are the satellite position coordinates at t_0. The transmitter and receiver coordinates are given by x_t, y_t, z_t and x_r, y_r, z_r respectively. For NAVSPASUR observations t_0, the time at which the satellite position and velocity are determined, is nominally the time of fence crossing.

Equation (A.3) provides a means to examine the relative information content of the Doppler frequency in light of the special geometry of the current NAVSPASUR system. Specifically, the sensitivity of the Doppler frequency to any satellite position or velocity component is given by the partial derivative of equation (A.3), taken with respect to that component. Taking the partial derivatives with respect to the velocity components, for example, yields the following equations:

$$\left.\frac{\partial \nu_D}{\partial v_x}\right|_{t=t_0} = -\frac{\nu_0}{c}\left[\frac{(x_s-x_t)}{|\vec{r}_{ts}|} + \frac{(x_s-x_r)}{|\vec{r}_{rs}|}\right] \tag{A.4}$$

$$\left.\frac{\partial \nu_D}{\partial v_y}\right|_{t=t_0} = -\frac{\nu_0}{c}\left[\frac{(y_s-y_t)}{|\vec{r}_{ts}|} + \frac{(y_s-y_r)}{|\vec{r}_{rs}|}\right] \tag{A.5}$$

$$\left.\frac{\partial \nu_D}{\partial v_z}\right|_{t=t_0} = -\frac{\nu_0}{c}\left[\frac{(z_s-z_t)}{|\vec{r}_{ts}|} + \frac{(z_s-z_r)}{|\vec{r}_{rs}|}\right]. \tag{A.6}$$

In deriving the above expressions we have evaluated the partial derivatives at time t_0 in order to simplify the resultant expressions and to clarify the functional dependences. Over the short period during which the satellite is illuminated, the time dependence of the partial derivatives is weak, so that our analysis is not seriously limited by this simplification.

In the coordinate system we have selected, both (z_s-z_t) and (z_s-z_r) are much smaller than $|\vec{r}_{ts}|$ and $|\vec{r}_{rs}|$ respectively, since all of the receiver and transmitter sites lie within the NAVSPASUR plane, and the satellite is only illuminated within a small distance of the plane. This implies that the Doppler frequency is relatively insensitive to v_z (*i.e.*, $\partial\nu_D/\partial v_z$ is small compared to

the remaining velocity partial derivatives). Note that the corresponding case is almost never true for the two in-plane velocities because of the bi-static nature of the NAVSPASUR system. In other words, if, for example, $(x_s-x_t) \approx (z_s-z_t)$, then $(x_s-x_r) >> (z_s-z_r)$, since the transmitter and receiver sites are well separated. For this reason the Doppler frequency will always contain significant information on both in-plane velocities. For typical distances encountered with NAVSPASUR observations, the two in-plane partial derivatives are of order 10 Hz per km/min, so that a Doppler measurement accuracy of ±10 Hz or better can constrain the in-plane velocity within the FPOD requirements. Typical values of the plane-normal partial derivatives, however, are of order 0.01 Hz per km/min, so that the plane-normal velocity cannot be adequately constrained for any reasonably achievable Doppler accuracy.

One can see from equation (A.6) that Doppler measurements from an OOPS receiver would provide enhanced sensitivity to the plane-normal velocity, since for stations well removed from the great circle plane (z_s-z_r) would be comparable to $|\vec{r}_{rs}|$.

In addition to *containing* information on the in-plane velocity components, the Doppler frequency also contains some information on all three position components. Taking the partial derivatives of equation (A.3) with respect to the position components and evaluating at time t_0, we have

$$\frac{\partial \nu_D}{\partial x_s}\bigg|_{t=t_0} = -\frac{\nu_0 \ V_x}{c}\left[\frac{1}{|\vec{r}_{ts}|} + \frac{1}{|\vec{r}_{rs}|}\right] \tag{A.7}$$

$$\frac{\partial \nu_D}{\partial y_s}\bigg|_{t=t_0} = -\frac{\nu_0 \ V_y}{c}\left[\frac{1}{|\vec{r}_{ts}|} + \frac{1}{|\vec{r}_{rs}|}\right] \tag{A.8}$$

$$\frac{\partial \nu_D}{\partial z_s}\bigg|_{t=t_0} = -\frac{\nu_0 \ V_z}{c}\left[\frac{1}{|\vec{r}_{ts}|} + \frac{1}{|\vec{r}_{rs}|}\right]. \tag{A.9}$$

The relative sensitivity of the Doppler to each of the position coordinates depends on the specifics of the satellite orbit, but, in general, the three velocity components are comparable, so that the sensitivity to each position coordinate is comparable. Typical values for the position partial derivatives are of order 1 Hz per km, so that Doppler information is not expected to

significantly improve upon the position determinations derived from the phase data.

In the case of chirp measurements, we have, taking the time derivative of equation (A.2),

$$\dot{\nu}_D = -\frac{\nu_0}{c}\left\{ \vec{a}_s \cdot [\hat{r}_{ts} + \hat{r}_{rs}] + \vec{v}_s \cdot \frac{d}{dt}[\hat{r}_{ts} + \hat{r}_{rs}] \right\} \tag{A.10}$$

where the satellite acceleration, $\vec{a}_s$, is given by

$$\vec{a}_s = -\frac{GM_\oplus}{|\vec{R}|^2}\hat{R}. \tag{A.11}$$

Here, $\vec{R}$ is the satellite position vector (earth-centered), G is the gravitational constant, and $M_\oplus$ is the mass of the earth. For typical values of the distances encountered with NAVSPASUR satellite observations, the first term of equation (A.10) is

$$\frac{GM_\oplus}{|\vec{R}|^2}\ \hat{R} \cdot [\hat{r}_{ts} + \hat{r}_{rs}]\ \frac{\nu_0}{c} < 7\ \text{Hz sec}^{-1}. \tag{A.12}$$

Since the satellite can only be detected if it is above the horizon of both the transmitter and receiver, both dot products in equation (A.12) are positive, so that the contribution of the acceleration term to the observed chirp is positive, and, as we show below, is a small part of the total chirp. Further, the contribution of this term depends only on the satellite position and may be calculated from the position, which is normally well determined by the data.

The second term of equation (A.10) reduces to

$$-\frac{\nu_0}{c}\vec{v}_s \cdot \frac{d}{dt}[\hat{r}_{ts} + \hat{r}_{rs}] = -\frac{\nu_0}{c}\left\{ \frac{1}{|\vec{r}_{ts}|}\left[|\vec{v}_s|^2 - (\vec{v}_s \cdot \hat{r}_{ts})^2 \right] + \frac{1}{|\vec{r}_{rs}|}\left[|\vec{v}_s|^2 - (\vec{v}_s \cdot \hat{r}_{rs})^2 \right] \right\}. \tag{A.13}$$

The terms in the square brackets in equation (A.13) are always greater than or equal to zero, since $|\vec{v}_s|$ is greater than or equal to both $|\vec{v}_s \cdot \hat{r}_{ts}|$ and $|\vec{v}_s \cdot \hat{r}_{rs}|$. Further, due to the bi-static nature of the system, if one of the terms in the square brackets is zero, the other is not. Therefore, the net chirp due to the second term in equation (A.10) is negative for all satellite passes. For typical NAVSPASUR satellite observations, the chirp contribution of the second term is between -20 and -100 Hz/sec.

As in the case of the Doppler frequency, we can examine the information content of the chirp by taking the partial derivatives of equation (A.10) with respect to the position and velocity components. Taking the velocity partials, again at time t_0, we have

$$\left.\frac{\partial \dot{\nu}_D}{\partial v_x}\right|_{t=t_0} = -\frac{\nu_0}{c}\left\{ \frac{2}{|\vec{r}_{ts}|}\left[v_x - (\vec{v}_s \cdot \hat{r}_{ts})\frac{(x_s - x_t)}{|\vec{r}_{ts}|}\right] + \frac{2}{|\vec{r}_{rs}|}\left[v_x - (\vec{v}_s \cdot \hat{r}_{rs})\frac{(x_s - x_r)}{|\vec{r}_{rs}|}\right]\right\} \qquad (A.14)$$

$$\left.\frac{\partial \dot{\nu}_D}{\partial v_y}\right|_{t=t_0} = -\frac{\nu_0}{c}\left\{ \frac{2}{|\vec{r}_{ts}|}\left[v_y - (\vec{v}_s \cdot \hat{r}_{ts})\frac{(y_s - y_t)}{|\vec{r}_{ts}|}\right] + \frac{2}{|\vec{r}_{rs}|}\left[v_y - (\vec{v}_s \cdot \hat{r}_{rs})\frac{(y_s - y_r)}{|\vec{r}_{rs}|}\right]\right\} \qquad (A.15)$$

$$\left.\frac{\partial \dot{\nu}_D}{\partial v_z}\right|_{t=t_0} = -\frac{\nu_0}{c}\left\{ \frac{2}{|\vec{r}_{ts}|}\left[v_z - (\vec{v}_s \cdot \hat{r}_{ts})\frac{(z_s - z_t)}{|\vec{r}_{ts}|}\right] + \frac{2}{|\vec{r}_{rs}|}\left[v_z - (\vec{v}_s \cdot \hat{r}_{rs})\frac{(z_s - z_r)}{|\vec{r}_{rs}|}\right]\right\}. \qquad (A.16)$$

Equations (A.14) - (A.16) demonstrate that the chirp will in general contain information on all three velocity components.

Because both $(z_s - z_t) << |\vec{r}_{ts}|$ and $(z_s - z_r) << |\vec{r}_{rs}|$, equation (A.16) is approximately

$$\frac{\partial \dot{\nu}_D}{\partial v_z} \approx -\frac{2\ v_z\ \nu_0}{c} \left[\frac{1}{|\vec{r}_{ts}|} + \frac{1}{|\vec{r}_{rs}|} \right]. \tag{A.17}$$

The sensitivity of the chirp to the plane-normal velocity is thus proportional to the plane-normal velocity. In the case of the in-plane components, the functional dependence of the partial derivatives is somewhat more complicated. However, one can deduce from an examination of equations (A.14) and (A.15) that the contribution of the in-plane terms is generally of the same order of magnitude as for the plane-normal term. This result is in contrast to the result obtained for the Doppler measurement, where the plane-normal velocity sensitivity is negligible. For typical NAVSPASUR observations, the partial derivatives of the chirp with respect to the velocity components are of order 0.05 Hz/second per km/min. In general, then, chirp measurements do not provide as stringent a constraint on the velocity as the Doppler measurements.

Summary

We have shown that, in general, both the Doppler and chirp measurements contain information on all satellite position and velocity components. Doppler measurements from in-plane receiving sites are far more sensitive to in-plane velocity than to plane-normal velocity. Given the typical magnitude of the Doppler partial derivatives with respect to velocity, we expect that Doppler measurement accuracies of ±10 Hz can provide in-plane velocity determinations of order ±1 km/min, much better than that attainable from direction cosine rates derived from the phase data. However, for the current system, the Doppler frequency cannot constrain plane-normal velocity nearly as well as the phase data. In order to achieve plane-normal velocity accuracies comparable to the in-plane velocity accuracy, either the target satellite must be illuminated well off the NAVSPASUR great circle plane, or out-of-plane receiver stations must be added, or both.

In the case of position information, Doppler measurements cannot substantially improve upon the accuracy attainable from the direction cosines derived from the phase data.

Finally, while chirp measurements yield in-plane and plane-normal velocity accuracies which are comparable, in neither case is the sensitivity sufficient to yield significant improvement over the accuracies available from either the phase or Doppler information.

```
C
C ***********************************************************************
C *                                                                     *
C *                           PROGRAM ORBITS                            *
C *                                                                     *
C ***********************************************************************
C
C  AUTHOR:        Dr. E. James Wadiak
C  DATE:          29-JAN-1988
C  LANGUAGE:      FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:          VX7770::SPACE:[WADIAK.LSQ.SIM.ORBITS]ORBITS.FOR
C
C  SUBROUTINES CALLED:  GEOTOGC.FOR
C
C  COMPILE INSTRUCTIONS:        FORTRAN ORBITS
C
C  LINK/LOAD INSTRUCTIONS:      LINK ORBITS
C
C  I/O UNIT ASSIGNMENTS:
C
C       FOR037 - SPACE:[WADIAK.LSQ.SIM]RECGC.POS
C
C  PROGRAM DESCRIPTION:
C
C    This program is intended to generate satellite positions and velocities
C  at the time of fence crossing for circular orbits.  The reference equations
C  are in FUNDAMENTALS OF ASTRODYNAMICS, page 82.  For convenience, the
C  perifocal <P> unit vector is taken to be in the direction of the satellite
C  at the time of fence crossing.  Then, nu, the true anomaly at epoch, is
C  zero, and the satellite's position is in the <P> direction and the velocity
C  is in the <Q> direction.
C
C    The inputs to the program are the orbital inclination, the altitude above
C  the ellipsoidal earth, and the longitude of fence crossing.  The altitudes
C  are contained in a DATA statement.  The longitudes are specified in the
C  DO LOOP limits.  The satellite position and velocity are calculated in
C  geocentric coordinates and are converted to NAVSPASUR great circle
C  coordinates for output.
C
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. READ in the rotation angles ALPHA and BETA, which are used to convert
C       from geocentric to great circle coordinates.  These are read from the
C       receiver great circle position file RECGC.POS to insure consistency
C       with the data simulation and least squares fitting programs.
C
C    2. OPEN the output file ORBITS.DAT.
C
C    3. DO, for longitudes between 240 and 285 degrees, in 5 degree increments:
C
C       3a. Find the latitude of the NAVSPASUR great circle at the given
C           longitude.  The nominal latitude is obtained from a cubic equation
C           fit to the positions of the receiving stations.  This  method
C           provides sufficient accuracy to properly position the satellite
C           within a few kilometers of the fence.
C
C       3b. DO, for the range of altitudes specified in the array ALT:
C
C           3d. Convert the satellite position, given in geodetic latitude,
C               longitude, and altitude above the geoid, to geocentric x,y,z
C               coordinates.
C
C           3e. Compute the satellite geocentric velocity components in a
C               non-rotating coordinate system.  The computation assumes a
```

```
C              circular orbit with an orbital inclination of 85 degrees and
C              a Keplerian gravitational potential.
C
C          3f. Compute the velocity in a rotating geocentric coordinate
C              system via the Coriolis Theorem.
C
C          3g. Convert the satellite position and velocity to NAVSPASUR great
C              circle coordinates.
C
C          3h. Force the satellite to lie in the great circle plane by setting
C              plane-normal coordinate to be equal to zero.  This small
C              correction is necessary to compensate for errors introduced by
C              the relatively imprecise manner in which the fence crossing
C              latitude was computed.
C
C          3i. WRITE the satellite altitude, longitude, orbital inclination,
C              position, and velocity to the output file.
C
C      3j. END of both DO loops.
C
C
C INPUTS   EXPLICIT:
C
C              FOR037 - ALPHA:  geocentric to great circle
C                       BETA:   rotation angles (in degrees).
C
C          IMPLICIT (via DATA statement):
C
C                     ALT(1-13) - altitudes for which orbits are to
C                                 be computed (in kilometers).
C
C                   (via assignment statement):
C
C                     INCL - orbital inclination (in degreea).
C
C OUTPUTS  EXPLICIT (via WRITE to FOR039):
C
C                     IALT - satellite altitude, in kilometers.
C                     INCL - satellite orbital inclination, in degrees.
C                     LONG - satellite longitude, in degrees.
C                     RS - satellite great circle position coordinates.
C                     VS - satellite great circle velocity coordinates.
C
C          IMPLICIT:   NONE
C
C MAJOR VARIABLES:
C
C              ALPHA - geocentric to great circle rotation angle #1.
C              ALT(1-13) - altitudes for which orbits are to computed (in km).
C              APERI - argument of the perigee.
C              CA - cos(ALPHA).
C              CAPERI - cas(APERI).
C              CB - cos(BETA).
C              CINC - cos(orbital inclination).
C              COMEGA - cas(OMEGA).
C              DLAT - latitude of fence crossing, in degrees (REAL*8).
C              DLONG - longitude of fence crossing, in degrees (REAL*8).
C              DTORAD - conversion factor: degrees to radians.
C              EARTHM - earth mass, in kilograms.
C              GRAV - universal gravitational constant, in MKS units.
C              IALT - satellite altitude, in kilometers.
C              INCL - satellite orbital inclination, in degrees.
C              LONG - satellite longitude, in degrees.
C              OMEGA - longitude of the ascending node.
C
C              R11:
```

```
C                      el...ts ... .e r    .on    ix f    onv    n b    on
C              R31:  the perifocal and geocentric coordinate systems.  The
C              R12:  choice of coordinate system is such that R13, R23, R33
C              R22:  are all zero.
C              R32:
C
C              REARTH - earth radius, in meters.
C              RELLIP - earth ellipticity (dimensionless).
C              RLAT - latitude of fence crossing, in radians.
C              RLONG - longitude of fence crossing, in degrees.
C              ROTRAT - earth angular velocity, in radians/sec.
C              RS - satellite great circle position coordinates.
C              RSAT - height of the satellite above geocenter, in meters.
C              SA - sin(ALPHA).
C              SAPERI - sin(APERI).
C              SB - sin(BETA).
C              SINC - sin(orbital inclination).
C              SOMEGA - sin(OMEGA).
C              VQ - satellite velocity, in meters/sec.
C              VS - satellite great circle velocity coordinates.
C
C
C
C  MODIFIED:
C
C
C*******************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ALT(13),RS(3),VS(3)
      CHARACTER*42 OUTFIL
      DATA ALT / 100.,200.,300.,500.,700.,1000.,
     +           1300.,1600.,2000.,2500.,3000.,3500.,4000. /
C
C  Enter constants.
C
      INCLUDE 'SPACE:[WADIAK.LSQ.SIM]CONSTANTS.FOR/LIST'
C
C  Specify the orbital inclination.  We will use 85 degrees for now.
C
      INCL = 85
      RRINC = FLOAT(INCL) * DTORAD
      SINC = DSIN(RRINC)
      CINC = DCOS(RRINC)
C
C  ALPHA and BETA are the rotation angles to be used to transform
C  to/from the great circle coordinate system. These are read from
C  the receiving antenna file to insure the same angles are
C  used throughout.
C
      READ(37,*) ALPHA,BETA
      CLOSE(37)
      DALPHA = ALPHA * DTORAD
      DBETA  = BETA  * DTORAD
      SA = DSIN(DALPHA)
      CA = DCOS(DALPHA)
      SB = DSIN(DBETA)
      CB = DCOS(DBETA)
C
C  Set the output file to ORBITS.DAT and open it.
C
      OUTFIL = 'SPACE:[WADIAK.LSQ.SIM.ORBITS]ORBITS.DAT'
      OPEN(UNIT=39,FILE=OUTFIL,STATUS='NEW')
C
C  Begin calculating orbits.  This is the loopback point.  Loop
C  first through longitude.  Find the latitude of fence crossing
C  for a given longitude using a cubic fit to the lat/long coordinates
```

```
C  of the NAVSPASUR receivers.
C
      DO LONG=240,285,5
        DLONG = FLOAT(LONG)
        DLAT = (-4.97133D-6 * DLONG**3) + (-2.31823D-4 * DLONG**2) +
     +         (1.11944 * DLONG) - 154.432
        RLONG = DLONG * DTORAD
        RLAT = DLAT * DTORAD
C
C  We will obtain the geocentric coordinates of the satellite from the
C  geodetic latitude, longitude and altitude via FUND. OF ASTR. page 98,
C  equations (2.8-7) and (2.8-8).  The first terms in equations (2.8-7)
C  depend only on latitude, so calculate them now.
C
        RADICL = DSQRT( 1 - RELLIP**2 * (DSIN(RLAT))**2 )
        ENX = REARTH / RADICL
        ENZ = ENX * ( 1. - RELLIP**2 )
C
C  Now loop through by altitude and calculate the satellite positions
C  and velocities.  Convert the altitudes to meters (given in kilometers
C  above!!).
        DO L=1,13
          IALT = NINT(ALT(L))
          H = ALT(L) * 1.D3
          RS(1) = ( ENX + H ) * DCOS(RLAT) * DCOS(RLONG)
          RS(2) = ( ENX + H ) * DCOS(RLAT) * DSIN(RLONG)
          RS(3) = ( ENZ + H ) * DSIN(RLAT)
          RSAT = DSQRT( RS(1)*RS(1) + RS(2)*RS(2) + RS(3)*RS(3) )
C
C  For an assumed circular orbit, the orbital velocity (which lies in
C  the adopted perifocal <Q> direction) is given by:
C
          VQ = DSQRT( GRAV * EARTHM / RSAT )
C
C  Use the satellite coordinates and the perifocal-to-geocentric
C  transformations (ref. FUND. of ASTR. p.82) to get the longitude of
C  the ascending node (OMEGA) and the argument of the perigee (APERI).
C  From these and VQ, the velocity components in the geocentric system
C  are calculated.  Finally, the geocentric R and V are transformed to
C  great circle coordinates and output.
C
C  First, get the angles.
C
          R11 = RS(1) / RSAT
          R21 = RS(2) / RSAT
          R31 = RS(3) / RSAT
          APERI = DASIN( R31 / SINC )
          CAPERI = DCOS(APERI)
          SAPERI = DSIN(APERI)
          A = CAPERI
          B = SAPERI * CINC
          COMEGA = ( R11 + R21 * B / A ) / (A + B * B / A )
          OMEGA = - DACOS(COMEGA)
          SOMEGA = DSIN(OMEGA)
C
C  Now, obtain the geocentric velocity components directly from the
C  angles and VQ.
C
          R12 = - COMEGA * SAPERI - SOMEGA * CAPERI * CINC
          R22 = - SOMEGA * SAPERI + COMEGA * CAPERI * CINC
          R32 = CAPERI * SINC
          VS(1) = R12 * VQ
          VS(2) = R22 * VQ
          VS(3) = R32 * VQ
C
C  All of the above has been done with respect to a NONROTATING earth.
```

```fortran
C  The geocentric X and Y velocities must be corrected to compensate
C  for the fact that the geocentric system we use is a ROTATING
C  coordinate system.
C
C  Use the Coriolis theorem:
C
C               VROT = VFIX - OMEGA X RSAT
C
C  where OMEGA X RSAT is the vector cross-product of the earth's angular
C  velocity and the satellite position vector.  Since the angular
C  velocity vector is parallel to the z axis, we have
C
C      - OMEGA X RSAT = (ROTRAT*RSY) <X> - (ROTRAT*RSX) <Y>
C
C  where <X>, <Y> are unit vectors in the X & Y directions.
C
C  The earth's angular velocity (fixed-star to fixed-star) is
C  1 revolution each 23 hours, 56 minutes.
C
            ROTRAT = TWOPI / (23. * 3600. + 56. * 60.)
            VXERTH = + ROTRAT * RS(2)
            VYERTH = - ROTRAT * RS(1)
            VS(1) = VS(1) + VXERTH
            VS(2) = VS(2) + VYERTH
C
C  Transform the position and velocity into great circle coordinates
C  and write to the output file.  Force the satellite into the NAVSPASUR
C  great circle plane by setting RS(3) = 0. (This is a small correction
C  necessary due to the manner in which we derive the latitude of fence
C  crossing).
C
            CALL GEOTOGC(RS,ALPHA,BETA)
            RS(3) = 0.0
            CALL GEOTOGC(VS,ALPHA,BETA)
            WRITE(39,401) IALT,LONG,INCL,RS,VS
          ENDDO
        ENDDO
 401    FORMAT(1X,3I5,/,3(1X,F12.2),/,3(1X,F12.2),/)
        STOP
        END
        INCLUDE  'SPACE:[WADIAK.LSQ.SIM.ORBITS]GEOTOGC.FOR/LIST'
```

```
      SUBROUTINE GEOTOGC(V,ALPHA,BETA)
C
C ***********************************************************************
C *                                                                     *
C *                         SUBROUTINE GEOTOGC                          *
C *                                                                     *
C ***********************************************************************
C
C  AUTHOR:       Dr. E. James Wadiak
C  DATE:         15-JAN-1988
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ.SIM.ORBITS]GEOTOGC.FOR
C
C  CALLING ROUTINE:     ORBITS.FOR
C
C  SUBROUTINES CALLED:  NONE
C
C  COMPILE INSTRUCTIONS:        via INCLUDE statement in calling routine.
C
C  LINK/LOAD INSTRUCTIONS:      via INCLUDE statement in calling routine.
C
C  PARENT PROGRAM:      ORBITS.FOR
C
C  PROGRAM DESCRIPTION:
C
C  This subroutine performs a double precision rotation of a 3-D position or
C  velocity vector V through angles ALPHA and BETA.
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. Calculate the sines and cosines of the relevant angles.
C
C    2. Use intermediate variables to hold the the vector components.
C
C    3. Use the two-angle rotation matrix to obtain the vector components
C       in the new (rotated) coordinate system.
C
C    4. RETURN the result to the calling program via the argument list.
C
C  INPUTS   EXPLICIT (via argumetns to CALL statement):
C
C               ALPHA - rotation angle #1 (west longitude), in degrees.
C               BETA - rotation angle #2 (north latitude), in degrees.
C               V(1-3) - x,y,z components of vector to be rotated.
C
C           IMPLICIT:   NONE
C
C  OUTPUTS  EXPLICIT (via arguments to CALL statement):
C
C               V(1-3) - rotated vector components in NAVSPASUR great circle
C                        coordinate system.
C
C           IMPLICIT:   NONE
C
C  MAJOR VARIABLES:
C
C               ALPHA - rotation angle #1 (west longitude), in degrees.
C               BETA - rotation angle #2 (north latitude), in degrees.
C               CA - cos(ALPHA).
C               CB - cos(BETA).
C               SA - sin(ALPHA).
C               SB - sin(BETA).
C               V(1-3) - On input: vector to be rotated.
C                        On output: rotated vector.
C
C               X:
```

```
C               Y: Intermediate variables to hold vector components.
C               Z:
C
C
C  MODIFIED:
C
C
C*****************************************************************************
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION V(3)
        SA = DSIND(ALPHA)
        CA = DCOSD(ALPHA)
        SB = DSIND(BETA)
        CB = DCOSD(BETA)
        X = V(1)
        Y = V(2)
        Z = V(3)
        V(1) = (CA*CB) * X + (SA*CB) * Y - (SB) * Z
        V(2) =  - (SA) * X +     (CA) * Y
        V(3) = (CA*SB) * X + (SA*SB) * Y + (CB) * Z
        RETURN
        END
```

```
C
C **********************************************************************
C *                                                                    *
C *                         PROGRAM SIMDAT                             *
C *                                                                    *
C **********************************************************************
C
C  AUTHOR:      Dr. E. James Wadiak
C  DATE:        12-FEB-1988
C  LANGUAGE:    FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:        VX7770::SPACE:[WADIAK.LSQ.SIM]SIMDAT.FOR
C
C  SUBROUTINES CALLED:          GNRTR.FOR, ERROR.FOR
C
C  COMPILE INSTRUCTIONS:        FORTRAN SIMDAT
C
C  LINK/LOAD INSTRUCTIONS:      LINK SIMDAT
C
C  PROGRAM DESCRIPTION:
C
C     This program is designed to generate simulation data for satellite orbital
C  passes through various points of the NAVSPASUR fence.  The program input is
C  a file of satellite positions and velocities at the time of fence crossing,
C  generated by the program ORBITS.FOR. The program output is a calibrated
C  phase difference data file in the format used for input to the multi-station
C  least-squares program.  One file is created for each satellite pass.
C  Gaussian errors are added to the phase difference data using the current
C  NAVSPASUR phase difference error model.
C
C     Receiver and transmitter positions and satellite positions and velocities
C  are input in NAVSPASUR great circle coordinates.  The internal calculations
C  are done in this coordinate system; however, to maintain consistency with
C  the *.DIF file format, the satellite position and velocity are rotated to
C  geocentric coordinates before output.
C
C  The included COMMON statement dimensions the arrays and passes most of the
C  needed variables to the subroutines.
C
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. READ in some control information, such as the number of transmitter and
C       receiver sites to process, the effective radar cross-section, the phase
C       quantization size, and the gain of the OOPS from FOR015 (SIMDAT.INP).
C
C    2. READ in the geocentric-to-great circle coordinate transformation
C       rotation angles, number of antennas at each receivinng site, number of
C       the reference antenna at each site, and positions for each antenna at
C       each of the sites from FOR037 (RECGC.POS).
C
C    3. READ in the transmitter positions from FOR038 (TRANGC.POS).
C
C    4. READ in the satellite altitude, longitude, orbital inclination, as well
C       as its position and velocity at the time of fence crossing, from a
C       control list FOR039 (*.DAT).  On end-of-file condition GOTO STOP.
C            **** THIS IS THE RETURN POINT FOR ADDITIONAL ORBITS ****
C
C    5. Set the output file data set name based on the OOPS gain, longitude,
C       and altitude.  OPEN the output file to FOR088.
C
C    6. Calculate the satellite position and velocity in geocentric coordinates.
C
C    7. DO, for each transmitter site:
C
C       7a. IF satellite is not above the horizon, go to next transmitter.
```

```
C
C      7b. DO, for each receiver site:
C
C          7c. IF satellite is not above horizon, go to next receiver.
C
C          7d. CALL subroutine GNRTR to actually generate the data.
C
C          7e. IF data scan is not empty, CALL subroutine ERROR to add
C              random errors to the phase difference data.
C
C          7f. Set the Doppler frequency and chirp to their values at the
C              time of the first data line, RSEC (calculated in GNRTR).
C
C          7g. Write the data scan to the output file FOR008 (*.DIF).
C
C    8. END both DO loops.  At this point the output file contains data scans
C       for each transmitter/receiver combination for which the satellite is
C       above the horizon and for which the received signal exceeds the cutoff.
C
C    9. CLOSE the current data file and GOTO step #4 to process the next orbit.
C
C
C  INPUTS   EXPLICIT:
C
C      FOR015 (SIMDAT.INP):
C             GNOOPS - OOPS receiver gain in dB.
C             NQNT - phase data quantization factor, in bits per rotation.
C             NREC - number of receiver sites to use.
C             NTRANS - number of transmitter sites to use.
C
C      FOR037 (RECGC.POS):
C             ALPHA - geocentric-to-great circle rotation angle #1.
C             BETA - geocentric-to-great circle rotation angle #2.
C             NANT(1-10) - number of receiving antennas at each rcvr site.
C             NREF(1-10) - number of the reference antenna at each rcvr site.
C             POS(10,12,3) - great circle coordinates of each antenna at
C                            each recceiving site.
C
C      FOR038 (TRANGC.POS):
C             TPOS(5,3) - great circle coordinates of each transmitter site.
C
C      FOR039 (*.DAT):
C             IALT - satellite altitude, kilometers.
C             INCL - satellite orbital inclination, in degrees.
C             LONG - satellite longitude, in degrees.
C             PSAT(1-3) - satellite geocentric x,y,z position, meters.
C             VSAT(1-3) - satellite geocentric x,y,z velocity, in meters/sec.
C
C          IMPLICIT:
C             (via COMMON statement):
C             DATA(11,55) - simulated phase difference data from GNRTR/ERROR.
C             DOPLR(1) - Doppler frequenccy at time RSEC from GNRTR.
C             DOPRAT - chirp at time RSEC from GNRTR.
C             NLINES - number of data lines from GNRTR.
C             RSEC - time of first data line from GNRTR.
C
C             (via CONSTANTS.FOR):
C             DTORAD - converts degrees to radians.
C             EARTHM - earth mass, in kilograms.
C             FTTOM - converts feet to meters.
C             FREQ - NAVSPASUR transmit frequency, in Hz.
C             GRAV - universal gravitational constant.
C             PI - value of pi.
C             REARTH - earth radius, in meters.
C             TINCR - NAVSPASUR data rate, in seconds.
C             TWOPI - 2 pi.
```

```
C              VLIGHT - velocity of light, in meters/second.
C              WVLN - NAVSPASUR transmit wavelength, in meters.
C
C
C  OUTPUTS  EXPLICIT:   FOR008 (*.DIF) - phase difference data scans for
C                                  each satellite pass.
C
C  MAJOR VARIABLES:
C
C              ALPHA - geocentric-to-great circle rotation angle #1.
C              BETA - geocentric-to-great circle rotation angle #2.
C              CA - cos(ALPHA).
C              CB - cos(BETA).
C              DATA(11,55) - simulated phase difference data.
C              DOPLR(1) - Doppler frequenccy at time RSEC.
C              DOPRAT - chirp at time RSEC.
C              DTORAD - converts degrees to radians.
C              EARTHM - earth mass, in kilograms.
C              ELLIM - satellite elevation limit for processing scans
C                      (currently set to 2 degrees in the program).
C              FTTOM - converts feet to meters.
C              FREQ - NAVSPASUR transmit frequency, in Hz.
C              GNOOPS - OOPS receiver gain in dB.
C              GRAV - universal gravitational constant.
C              IALT - satellite altitude, kilometers.
C              INCL - satellite orbital inclination, in degrees.
C              ISAT - satellite number (used to hold orbital inclination).
C              ISTA - NAVSPASUR receiver station number.
C              LONG - satellite longitude, in degrees.
C              NANT(1-10) - number of receiving antennas at each rcvr site.
C              NLINES - number of data lines from GNRTR.
C              NQNT - phase data quantization factor, in bits per rotation.
C              NREC - number of receiver sites to use.
C              NREF(1-10) - number of the reference antenna at each rcvr site.
C              NTRANS - number of transmitter siytes to use.
C              PGEO(1-3) - satellite position in geocentric coordinates.
C              PI - value of pi.
C              POS(10,12,3) - great circle coordinates of each antenna at
C                             each recceiving site.
C              PSAT(1-3) - satellite geocentric x,y,z position, meters.
C              REARTH - earth radius, in meters.
C              RSEC - time of first data line from GNRTR.
C
C              RSX:
C              RSY: receiver-satellite x,y,z distances.
C              RSZ:
C
C              SA - sin(ALPHA).
C              SB _ sin(BETA).
C              TINCR - NAVSPASUR data rate, in seconds.
C              TPOS(5,3) - great circle coordinates of each transmitter site.
C              TPRED - predicted time of fence crossing (used to hold radar
C                      cross section information).
C
C              TSX:
C              TSY:  transmitter-satellite x,y,z distances.
C              TSZ:
C
C              TWOPI - 2 pi.
C              VGEO(1-3) - satellite velocity in geocentric coordinates.
C              VLIGHT - velocity of light, in meters/second.
C              VSAT(1-3) - satellite geocentric x,y,z velocity, in meters/sec.
C              WVLN - NAVSPASUR transmit wavelength, in meters.
C
C
C  FORMAT statement labels used:
```

```
      40     2,40     4,40     6
C
C  MODIFIED:
C
C
C***********************************************************************
        INCLUDE '[WADIAK.LSQ.SIM]SIMDAT.CMN/LIST'
        CHARACTER*42 OUTFIL
C
C  Enter constants.
C
        INCLUDE 'SPACE:[WADIAK.LSQ.SIM]CONSTANTS.FOR/LIST'
C
C  Set a minimum elevation limit of 2 degrees for processing
C  an observation.
C
        ELLIM = 2.0 * ( TWOPI / 360. )
        SINLIM = DSIN(ELLIM)
C
C  Get some control data.  NREC is the number of receiving sites and
C  NTRANS the number of transmitting stations.   Presently, NREC <= 10
C  and NTRANS <= 5.  SIZE is the satellite radar cross-section in m**2,
C  and NQNT is the phase quantization factor.  GNOOPS is the OOPS
C  primary beam peak power gain in db.
C
        READ(15,*) NREC,NTRANS,SIZE,NQNT
        READ(15,*) GNOOPS
        CLOSE(15)
        NSITES = NREC + NTRANS
C
C  Input the geocentric to great circle rotation angles from RECGC.POS.
C
        READ(37,*) ALPHA,BETA
        ALPHA = ALPHA * DTORAD
        BETA  = BETA  * DTORAD
        SA = DSIN(ALPHA)
        CA = DCOS(ALPHA)
        SB = DSIN(BETA)
        CB = DCOS(BETA)
C
C  Input the number of antennas at each of the receiver sites and the
C  antenna number of the reference antenna at each site.
C
        READ(37,*) (NANT(J),J=1,NREC)
        READ(37,*) (NREF(J),J=1,NREC)
C
C  Input receiving station and antenna positions for NREC receivers.
C  These are in great circle coordinates.
C
        DO ISTA=1,NREC
          DO NNT=1,NANT(ISTA)
            READ(37,*) (POS(ISTA,NNT,J),J=1,3)
          ENDDO
        ENDDO
        CLOSE(37)
C
C  Read in the positions for the transmitting stations from TRANS.POS.
C  These are also in great circle coordinates.
C
        READ(38,*)
        DO ITR=1,NTRANS
          READ(38,*) (TPOS(ITR,J),J=1,3)
        ENDDO
        CLOSE(38)
C
C  Read in the satellite altitude, longitude of fence crossing,
```

```
C  orbital inclination, and the position and velocity, at SENCE
C  crossing in great circle coordinates.  The data set name will
C  contain the longitude and altitude information.  We will use
C  TPRED, the epoch at crossing, and ISAT, the satellite number,
C  to carry the satellite size (m**2) and orbital inclination
C  (degrees), respectively.
C
C  ********************************************************
C  ****  This is the loopback point for multiple orbits.  ****
C  ********************************************************
C
 10     READ(39,*,END=999) IALT,LONG,INCL
        READ(39,*) PSAT
        READ(39,*) VSAT
        READ(39,*)
        TPRED = SIZE
        ISAT = INCL
C
C  Set the output data set name and open the file.
C
        IGN = NINT(GNOOPS)
        OUTFIL = '
        IF(IALT.LT.1000) THEN
            WRITE(OUTFIL,404) IGN,LONG,IALT
          ELSE
            IF(IALT.LT.10000) THEN
                WRITE(OUTFIL,405) IGN,LONG,IALT
              ELSE
                WRITE(OUTFIL,406) IGN,LONG,IALT
            ENDIF
        ENDIF
        OPEN(UNIT=8,FILE=OUTFIL,STATUS='NEW')
 404    FORMAT('SPACE:[WADIAK.LSQ.DATA]G',I2,'L',I3,'A',I3,'.DIF')
 405    FORMAT('SPACE:[WADIAK.LSQ.DATA]G',I2,'L',I3,'A',I4,'.DIF')
 406    FORMAT('SPACE:[WADIAK.LSQ.DATA]G',I2,'L',I3,'A',I5,'.DIF')
C
C  Calculate the satellite position and velocity in geocentric
C  coordinates in the arrays PGEO and VGEO.  These will be the
C  output coordinates for the *.DIF file.
C
        PX = PSAT(1)
        PY = PSAT(2)
        PZ = PSAT(3)
        VX = VSAT(1)
        VY = VSAT(2)
        VZ = VSAT(3)
        PGEO(1) = (CB*CA) * PX - (SA) * PY + (CA*SB) * PZ
        PGEO(2) = (CB*SA) * PX + (CA) * PY + (SA*SB) * PZ
        PGEO(3) =  - (SB) * PX +  0.0 * PY +    (CB) * PZ
        VGEO(1) = (CB*CA) * VX - (SA) * VY + (CA*SB) * VZ
        VGEO(2) = (CB*SA) * VX + (CA) * VY + (SA*SB) * VZ
        VGEO(3) =  - (SB) * VX +  0.0 * VY +    (CB) * VZ
C
C  Ready to generate the observational data.  The approach will be to
C  loop through, first by transmitter and then by receiver.  For each
C  transmitter, check to see if the satellite is above the elevation
C  limit.  If so, check each receiver in turn to see if the satellite
C  is above the elevation limit.  If so, generate a scan for that
C  transmitter/receiver combination.
C
        DO ITR=1,NTRANS
C
C
C  Transmitter elevation test.  For satellites near the great circle,
C  SIN(elevation angle) is, to sufficient accuracy, the dot product
C  <T>dot<TS>, where <T> is the transmitter position unit vector in
```

```
.   eat    :le    din.    and    . is    uni    cto:    m t
C  transmitter to the satelitte.
C
         TX = TPOS(ITR,1)
         TY = TPOS(ITR,2)
         TZ = TPOS(ITR,3)
         DTRAN = DSQRT( TX*TX + TY*TY + TZ*TZ )
         PX = PSAT(1)
         PY = PSAT(2)
         PZ = PSAT(3)
         DSAT = DSQRT( PX*PX + PY*PY + PZ*PZ )
         TSX = PX - TX
         TSY = PY - TY
         TSZ = PZ - TZ
         DTS = DSQRT( TSX*TSX + TSY*TSY + TSZ*TSZ )
         SINSAT = ( TX*TSX + TY*TSY + TZ*TSZ ) / ( DTRAN * DTS )
         IF(SINSAT.LT.SINLIM) GOTO 105
         DO ISTA=1,NREC
C
C  Receiver elevation test.  Same approach as above.
C
           RX = POS(ISTA,NREF(ISTA),1)
           RY = POS(ISTA,NREF(ISTA),2)
           RZ = POS(ISTA,NREF(ISTA),3)
           DREC = DSQRT( RX*RX + RY*RY + RZ*RZ )
           RSX = PX - RX
           RSY = PY - RY
           RSZ = PZ - RZ
           DRS = DSQRT( RSX*RSX + RSY*RSY + RSZ*RSZ )
           SINSAT = ( RX*RSX + RY*RSY + RZ*RSZ ) / ( DREC*DRS )
           IF(SINSAT.LT.SINLIM) GOTO 106
C
C   If the satellite passes the elevation tests, a data scan will be
C  generated.  All of the calculations are done in subroutine GNRTR.
C  The arguments specify the transmitter (ITR) and receiver (ISTA).
C  There will be NLINES*NANT(ISTA) phase differences returned
C  in the DATA array via the common block.
C
           CALL GNRTR(ITR,ISTA)
C
C  If there are data lines in the scan, add amplitude-dependent
C  Gaussian errors to the phase differences and write the output
C  to file SIMDAT.DIF.
C
           IF(NLINES.GT.0) THEN
             CALL ERROR(ISTA)
C
C  Set IDOP to the doppler frequency at time RSEC.
C  Convert the transmitter # to the NAVSPASUR convention (7,8,9).
C
             IDOP = NINT(DOPLR(1))
             ITRAN = ITR + 6
C
C  Write the header.
C
             WRITE(8,401) ISAT,ISTA,ITRAN,IDOP,RSEC,TPRED,NLINES,
     +                    DOPRAT,IALT,LONG
 401         FORMAT(1X,4I7,2F10.3,3X,I4,2X,F8.1,3X,2I5)
             WRITE(8,402) PGEO,VGEO
 402         FORMAT(1X,6F13.2)
C
C  To maintain consistency with the *.DIF file structure, only
C  NANT-1 phase differences are written to the output file.
C
             NREFM1 = NREF(ISTA) - 1
             NREFP1 = NREF(ISTA) + 1
```

```
            DO     1,NI
              WRITE(8,403) IAMP(II), (DATA(II,JJ),JJ=1,NREFM1),
     +                   (DATA(II,JJ),JJ=NREFP1,NANT(ISTA))
            ENDDO
403         FORMAT(1X,I3,11F7.3)
            WRITE(8,*)
          ENDIF
106     ENDDO
105   ENDDO
      GOTO 10
999   STOP
      END
      INCLUDE '[WADIAK.LSQ.SIM]ERROR.FOR/LIST'
      INCLUDE '[WADIAK.LSQ.SIM]GNRTR.FOR/LIST'
```

```
C     ***     ****     ****     ***     ***     ****     ****     ***     ***     ****
C*                                                                                  *
C*                              COMMON BLOCK SIMDAT                                 *
C*                                                                                  *
C***********************************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON NREC,NTRANS,NLINES,NSITES,NQNT,NLMAX,IX,IY,IDOP,
     +  NANT(10),NREF(10),NSIT(100),IAMP(55),
     +  TPRED,TINCR,CA,CB,SA,SB,T,FOBS,SIZE,ELLIM,TWOPI,VLIGHT,
     +  TREC,FREQ,RSEC,TX,TY,TZ,RX,RY,RZ,DOPRAT,DTORAD,GNOOPS,
     +  DOPLR(2),PGEO(3),PSAT(3),VSAT(3),VGEO(3),
     +  POS(10,12,3),TPOS(5,3),
     +  PHASE(12),AIONO(15),ANEUT(15),CLOCK(15),AMPLT(13),
     +  DATA(55,12)
```

```
C*********************************************************************
C*                                                                   *
C*                 PROGRAM SEGMENT CONSTANTS.FOR                     *
C*                                                                   *
C*********************************************************************
C
C  This program segment contains a number of physical and system constants
C  which are used throughout the NAVSPASUR analysis programs.
C  The constants are:
C
C             DTORAD - converts degrees to radians.
C             EARTHM - earth mass, in kilograms.
C             FTTOM - converts feet to meters.
C             FREQ - NAVSPASUR transmit frequency, in Hz.
C             GRAV - universal gravitational constant.
C             REARTH - earth radius, in meters.
C             RELLIP - ellipticity of the earth (dimensionless).
C             TINCR - NAVSPASUR data rate, in seconds.
C             VLIGHT - velocity of light, in meters/second.
C             WVLN - NAVSPASUR transmit wavelength, in meters.
C
      PI = 3.1415926536
      TWOPI = 2 * PI
      DTORAD = TWOPI / 360.
      FTTOM = 0.3048
      GRAV = 6.67D-11
      EARTHM = 5.976D+24
      REARTH = 6378135.0
      RELLIP = 0.08182
      VLIGHT = 2.997925D+8
      FREQ = 216.98D6
      WVLN = VLIGHT / FREQ
      TINCR=1D0/54.98D0
```

```
C
C ***********************************************************************
C *                                                                     *
C *                          SUBROUTINE GNRTR                           *
C *                                                                     *
C ***********************************************************************
C
C  AUTHOR:        Dr. E. James Wadiak
C  DATE:          12-FEB-1988
C  LANGUAGE:      FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:          VX7770::SPACE:[WADIAK.LSQ.SIM]GNRTR.FOR
C
C  CALLING ROUTINE:      SIMDAT.FOR
C
C  SUBROUTINES CALLED:   BEAM.FOR
C
C  COMPILE INSTRUCTIONS:         via INCLUDE statement in SIMDAT.FOR
C
C  LINK/LOAD INSTRUCTIONS:       via INCLUDE statement in SIMDAT.FOR
C
C  PARENT PROGRAM:       SIMDAT.FOR
C
C  PROGRAM DESCRIPTION:
C
C     This subroutine calculates the expected phase differences in a data
C  simulation.
C  ITR identifies the transmitter and ISTA the receiving station.
C  Everything else is passed by the common.  All phases are calculated
C  w.r.t. the satellite initially and later converted to differential
C  phases.
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C  INPUTS   EXPLICIT:
C
C           IMPLICIT:
C
C  OUTPUTS  EXPLICIT:
C
C           IMPLICIT:
C
C  MAJOR VARIABLES:
C
C  MODIFIED:
C
C
C***********************************************************************
        INCLUDE '[WADIAK.LSQ.SIM]SIMDAT.CMN'
C
C   Loop over NLINES data lines and NANT(ISTA) receiving antennas.
C
      NLINES = 0
      KK = 0
C
C  Find a maximum of 55 data lines with amplitudes above the cutoff.
C  Begin the calculations TPRED - 40*TINCR seconds.  If 55 lines are
C  not found after 110 tries (2 seconds), give up.
C
      DO WHILE (KK.LT.110.AND.NLINES.LT.55)
        KK = KK + 1
        T = ( KK - 41 ) * TINCR
C
C  Calculate the received signal strength at time T.  First step
C  is to get the distances from the satellite to the transmitter and
C  the receiver and the E-W and N-S angles for both transmitter and
```

```
.  .scel ...
C
C  X,Y,Z give the satellite position at time T.
C
       X = PSAT(1) + (T)*VSAT(1)
       Y = PSAT(2) + (T)*VSAT(2)
       Z = PSAT(3) + (T)*VSAT(3)
       DS = DSQRT( X*X + Y*Y + Z*Z )
C
C  TSX,TSY,TSZ give the satellite position relative to the transmitter.
C  These change with time and must be recalculated at each time step.
C  TX,TY,TZ  are the transmitter coordinates and are obviously constant.
C  These are passed from the calling program via the COMMON.
C
       TSX=X - TX
       TSY=Y - TY
       TSZ=Z - TZ
       DTS = DSQRT( TSX*TSX + TSY*TSY + TSZ*TSZ )
C
C  Calculate the transmitter-satellite angles.  To sufficient accuracy,
C  the E-W angle is ARCCOS(<T>dot<TS>), where <T> is the transmitter
C  position unit vector and <TS> is the unit vector from the transmitter
C  to the satellite.
C
C
       THETAT = DASIN( TSZ / DTS )
       DT = DSQRT( TX*TX + TY*TY + TZ*TZ )
       TDOTS = ( TX*TSX + TY*TSY + TZ*TSZ ) / ( DT * DTS )
       PHIT = DACOS( TDOTS )
C
C  Repeat the above for the receiver.
C
       RSX = X - RX
       RSY = Y - RY
       RSZ = Z - RZ
       DRS = DSQRT( RSX*RSX + RSY*RSY + RSZ*RSZ )
       THETAR = DASIN( RSZ / DRS )
       DR = DSQRT( RX*RX + RY*RY + RZ*RZ )
       RDOTS = ( RX*RSX + RY*RSY + RZ*RSZ ) / ( DR * DRS )
       PHIR = DACOS( RDOTS )
C
C  Call subroutine BEAM to calculate the amplitude of the received
C  signal.  AMPL is the returned amplitude (W/m**2).
C
       CALL BEAM(ITR,ISTA,THETAT,PHIT,DTS,THETAR,PHIR,DRS,
     +           SIZE,GNOOPS,AMPL)
       IAMPT = NINT( DABS(AMPL) )
C
C  Test to see if the amplitude is greater than the minimum cutoff.  If
C  so, generate a data line.
C
       IF(IAMPT.LE.152) THEN
           NLINES = NLINES + 1
           IAMP(NLINES) = IAMPT
C
C  If this is the first data line to be saved, set RSEC, the scan start
C  time and calculate the doppler frequency and doppler rate.
C
           IF(NLINES.EQ.1) RSEC = TPRED + T - 0.5 * TINCR
           IF(NLINES.EQ.1.OR.NLINES.EQ.2) THEN
             TSDOTV = ( TSX*VSAT(1) + TSY*VSAT(2) + TSZ*VSAT(3) ) / DTS
             RSDOTV = ( RSX*VSAT(1) + RSY*VSAT(2) + RSZ*VSAT(3) ) / DRS
             FREQ1 = FREQ  * ( 1. - TSDOTV / VLIGHT )
             FREQ2 = FREQ1 * ( 1. - RSDOTV / VLIGHT )
             DOPLR(NLINES) = FREQ2 - FREQ
           ENDIF
```

```
          IF(I....S.E... DOP.... = (D.....(2)   ..PLR.... / T.....
C
C  Calculate NANT(ISTA) phase shifts.
C
          DO III=1,NANT(ISTA)
            WVL = VLIGHT/FREQ2
            DRSX = X - POS(ISTA,III,1)
            DRSY = Y - POS(ISTA,III,2)
            DRSZ = Z - POS(ISTA,III,3)
            PATH = DSQRT( DRSX*DRSX + DRSY*DRSY + DRSZ*DRSZ )
            RTNS = PATH / WVL
            PHASE(III) = DMOD(RTNS,1D0)
C
C  Quantize the phases in units of 1/NQNT rotations.
C
            PHASE(III) = FLOAT( NINT(PHASE(III) * NQNT) ) / NQNT
          ENDDO
C
C  Fill the DATA array with differenced phases between -0.5 and +0.5
C  rotations.
C
          DO KKK=1,NANT(ISTA)
            DATA(NLINES,KKK) = -( PHASE(KKK) - PHASE(NREF(ISTA)) )
            DO WHILE (DATA(NLINES,KKK).LT.-0.5)
              DATA(NLINES,KKK) = DATA(NLINES,KKK) + 1.0
            ENDDO
            DO WHILE (DATA(NLINES,KKK).GT.+0.5)
               DATA(NLINES,KKK) = DATA(NLINES,KKK) - 1.0
            ENDDO
          ENDDO
C
C  If the amplitude drops BACK below -160 db after having exceeded this
C  this value, the satellite is passing out of the fence and we want to
C  end the scan.
C
        ELSE
          IF(NLINES.GT.0) RETURN
        ENDIF
      ENDDO
      RETURN
      END
      INCLUDE '[WADIAK.LSQ.SIM]BEAM.FOR/LIST'
```

```
      SU[illegible]INE [illegible]M (1[illegible],[illegible]STA,[illegible]AT,[illegible],DTS,[illegible]TAR,[illegible],DR[illegible],
     +                  SIZE,GNOOPS,AMPL)
C
C ***************************************************************************
C *                                                                         *
C *                          SUBROUTINE BEAM                                *
C *                                                                         *
C ***************************************************************************
C
C  AUTHOR:        Dr. E. James Wadiak
C  DATE:          12-FEB-1988
C  LANGUAGE:      FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:          VX7770::SPACE:[WADIAK.LSQ.SIM]BEAM.FOR
C
C  CALLING ROUTINE:      GNRTR.FOR
C
C  SUBROUTINES CALLED:  NONE
C
C  COMPILE INSTRUCTIONS:         via INCLUDE statement in calling routine.
C
C  LINK/LOAD INSTRUCTIONS:       via INCLUDE statement in calling routine.
C
C  PARENT PROGRAM:      SIMDAT.FOR
C
C  PROGRAM DESCRIPTION:
C
C  This subroutine will calculate the received signal strength for the given
C  transmitter/receiver pair.  The power at the satellite is explicitly
C  calculated based on the transmitter power and beam pattern and the distance
C  to the satellite.  The received signal strength is calculated from the power
C  at the satellite, the radar cross-section, the receiver beam pattern, and
C  the satellite-receiver distance.
C
C  The transmitter E-W patterns are modelled using a ninth-order polynomial fit
C  to Dr. Steven Berg's calculated beam pattern for an inverted-V dipole above
C  a finite wire-grid ground screen.  The Kickapoo N-S beam pattern was assumed
C  to be a Gaussian of 0.042 degrees half-power beam width in the far field.  A
C  near field correction term is also included in the Kickapoo beam model.  The
C  N-S beam patterns of Gila River and Jordan Lake are modelled as slot antennas
C  of the appropriate length, with all sidelobes truncated.
C
C  The receiver beam patterns are modelled as slot antennas in the N-S direction
C  and dipoles above an infinite ground screen in the E-W direction.
C
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. Calculate the transmitter E-W antenna gain in the direction of the
C       target satellite.
C
C    2. IF the transmitter is Kickapoo, use the Gaussian FWHP formula with a
C       near field correction term for the N-S gain.  Calculate the normal-
C       ization of the Gaussian.
C
C       2a. ELSE, use a slot antenna formula for the transmitter N-S gain.
C
C    3. Calculate the power at the satellite based on the transmitter power,
C       antenna gain in the direction of the satellite, and transmitter-to-
C       satellite distance.
C
C    4. IF the receiver is not an OOPS receiver, THEN
C
C       5a. Calculate the receiver N-S, E-W gains based on the satellite
C           position.
C
C    5. ELSE, set the receiver gain to the arbitrarily specified gain GNOOPS.
```

```
C
C   6. Calculate the power density at the receiver based on the power at the
C      satellite, the receiver antenna gain in the direction of the satellite,
C      and the satellite-to receiver distance.
C
C   7. Calculate the received power from the power density at the receiver and
C      the effective area of the receiving antenna.  Convert to dBm.
C
C   8. RETURN to the calling routine.
C
C
C INPUTS   EXPLICIT (arguments to CALL statement):
C
C                       DRS - satellite-receiver distance in meters.
C                       DTS - satellite-transmitter distance in meters.
C                       GNOOPS - OOPS receiver gain in dB.
C                       ISTA - receiver ID number.
C                       ITR - transmitter ID number.
C                       PHIR - receiver E-W angle to the satellite.
C                       PHIT - transmitter E-W angle to the satellite.
C                       SIZE - satellite radar cross-section in meters**2.
C                       THETAR - receiver N-S angle to the satellite.
C                       THETAT - transmitter N-S angle to the satellite.
C
C
C          IMPLICIT (in DATA statements):
C
C                       C(10) - array containing the coefficients of the
C                               polynomial fit to the E-W transmitter pattern.
C                       DNORM - receiver E-W gain normalization.
C                       EWNORM - transmitter E-W gain normalization.
C                       H - height of receiver dipoles above infinite ground
C                           screen, in meters.
C                       POWER(6) - transmitter output powers in Watts.
C                       RLNGTH(10) - receiver antenna N-S lengths in meters.
C                       RNORM(10) - receiver gain normalizations
C                                   (E-W, N-S combined).
C                       TLNGTH(6) - transmitter antenna N-S lengths in meters.
C                       TNORM(6) - transmitter N-S gain normalizations.
C
C OUTPUTS  EXPLICIT (arguments to CALL statement):
C
C                       AMPL - received amplitude in dBm.
C
C          IMPLICIT:    NONE
C
C MAJOR VARIABLES:
C
C                 AMPL - received amplitude in dBm.
C                 AMPREC - intermediate variable used to calculate receiver gain.
C                 AMPSAT - relected power from the satellite.
C                 C(10) - array containing the coefficients of the
C                         polynomial fit to the E-W transmitter pattern.
C                 DNORM - receiver E-W gain normalization.
C                 DRS - satellite-receiver distance in meters.
C                 DTS - satellite-transmitter distance in meters.
C                 EWNORM - transmitter E-W gain normalization.
C                 FWIDTH - Kickapoo N-S full width to half power.
C                 GNOOPS - OOPS receiver gain in dB.
C                 H - height of receiver dipoles above infinite ground screen.
C                 ISTA - receiver ID number.
C                 ITR - transmitter ID number.
C                 PHIR - receiver E-W angle to the satellite.
C                 PHIT - transmitter E-W angle to the satellite.
C                 POWER(6) - transmitter output powers in Watts.
C                 REFF - effective area of the receiving antenna.
C
```

```
C               REW - receiver E-W power gain.
C               RGAIN - receiver power gain in the direction of the satellite.
C               RLNGTH(10) - receiver antenna N-S lengths in meters.
C               RNORM(10) - receiver gain normalizations (E-W, N-S combined).
C               RNS - receiver N-S power gain.
C               SIGMA - Kickapoo N-S Gaussian dispersion.
C               SIZE - satellite radar cross-section in meters**2.
C               TEW - transmitter E-W power gain.
C               TGAIN - transmitter power gain in the direction of satellite.
C               THETAR - receiver N-S angle to the satellite.
C               THETAT - transmitter N-S angle to the satellite.
C               TLNGTH(6) - transmitter antenna N-S lengths in meters.
C               TNORM(6) - transmitter N-S gain normalizations.
C               TNS - transmitter N-S power gain.
C
C
C  MODIFIED:
C
C
C***********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 POWER(6),C(10),TLNGTH(6),TNORM(6),RLNGTH(10),RNORM(10)
C
C  Array C contains the coefficients of a 9th order polynomial fit to the
C  transmitter E-W beam (voltage) pattern calculated by Dr. Steven Berg.
C  The transmitter and receiver N-S and E-W normalizations were obtained by
C  integerating the gain functions over the upper half-plane and setting
C  the integrals to unity gain.
C
      DATA C / 66.27,13.41,-331.2,104.9,3157.,-8025.,8807.,
     +         -5061.,1499.,-181.7 /
      DATA EWNORM, DNORM, H / 7.565D3, 8.687, 0.28 /
      DATA POWER / 8.1D5, 2*4.5D4, 3*0.0 /
      DATA RLNGTH / 122.,732.,2*122.,732.,5*122. /
      DATA RNORM / 1.02D-2,1.70D-3,2*1.02D-2,1.70D-3,1.02D-2,
     +             20.,3.,3.,3. /
      DATA TLNGTH / 3249.,500.,315.,3*500. /
      DATA TNORM / 2.14D-4,2.63D-3,4.16D-3,3*2.63D-3 /
C
C  Enter constants.
C
      INCLUDE 'SPACE:[WADIAK.LSQ.SIM]CONSTANTS.FOR/LIST'
C
C  Calculate the signal strength at the satellite.  The E-W transmitter
C  beam has been fit to an 9th order polynomial.  For the coastal
C  transmitters (GILA RIVER and JORDAN LAKE) the N-S patterns have been
C  fit to slots of the appropriate lengths.  For KICKAPOO, the N-S beam
C  has been modelled as a Gaussian of FWHP 0.042 degrees in the far
C  field.  A near field correction term has been added to the Gaussian
C  FWHP.  The correction has been modelled as an additive term which is
C  proportional to the antenna array length and inversely proportional
C  to the TX-satellite distance.  The factor of 3 was determined
C  empirically by comparison of real and simulated data.
C
      TEW = 0.0
      DO I=10,1,-1
        TEW = (TEW * DABS(PHIT)) + C(I)
      ENDDO
C
C  Convert to power.
C
      TEW = TEW * TEW
C
C  If ITR = 1 then use KICKAPOO N-S expression and calculate
C  the correct normalization based on apparent beam FWHP.
```

```
C
        IF(ITR.EQ.1) THEN
            FWIDTH = (0.042 * DTORAD) + TLNGTH(ITR) / (3 * DTS)
            SIGMA = FWIDTH / 2.355
            TNS = DEXP( -0.5 * ( THETAT / SIGMA )**2 )
            TNORM(1) =  SIGMA * DSQRT( 2 * PI )
        ENDIF
C
C  Use a slot antenna expression for the coastal transmitters.
C
        IF(ITR.EQ.2 .OR. ITR.EQ.3) THEN
            BT = (PI * TLNGTH(ITR) / WVLN) * DSIN(THETAT)
C
C  Calculate the unnormalized N-S gain for the slot antenna.  If the satellite
C  is beyond the first null, set thhe gain to zero (i.e., kill the sidelobes).
C  If BT is small, sin(BT) = BT.  In this case, set the gain equal to 1 to
C  avoid possible zero divides.
C
            IF(DABS(BT).GE.PI) THEN
                TNS = 0.0
              ELSE
                IF(DABS(BT).LE.1.0D-2) THEN
                    TNS = 1.0
                  ELSE
                    TNS = ( DSIN(BT) / BT )**2
                ENDIF
            ENDIF
        ENDIF
C
C  Calculate the normalized transmittter power gain in the ddirection of the
C  satellite.
C
        TGAIN = ( TEW * TNS ) / ( EWNORM * TNORM(ITR) )
C
C  Calculate the reflected power from the satellite.
C
        AMPSAT = POWER(ITR) * TGAIN * SIZE / (4*PI*DTS*DTS)
C
C  Calculate the power density (dBm/m**2) at the receiving site.  If the
C  receiver is one of the OOPS receivers, use an arbitrary normalization and
C  isotropic gain.  For in-plane stations, the assumed receiver antenna
C  pattern is a sinc function N-S and a dipole above an infinite ground screen
C  E-W.  The real receiver dipoles are mounted 0.322 wavelengths above the
C  finite ground screen.  In order to get the E-W FWHP to agree with the
C  design spec of 128 degrees, we take the height H above the infinite
C  ground screen to be 0.28 wavelengths.
C
        IF(ISTA.LE.6) THEN
            REW = ( 1. - DCOS( 4 * PI * H * DCOS(PHIR) ))
C
C  Model the N-S pattern as a sinc function (i.e., a slot antenna).  The sinc
C  function has terrible sidelobes - set the gain to zero past the first null
C  to correct this problem.
C
            BR = (PI * RLNGTH(ISTA) / WVLN) * DSIN(THETAR)
            IF(DABS(BR).GE.PI) THEN
                RNS = 0.0
              ELSE
                IF(DABS(BR).LE.1.0D-2) THEN
                    RNS = 1.0
                  ELSE
                    RNS = ( DSIN(BR) / BR )**2
                ENDIF
            ENDIF
C
C  The factor of 4*PI in RGAIN comes from the requirement that the
```

```
C  teg   over   ce c   uni   in   na   *PI.   e f
C  of 2 is included raise the receiver antenna gains 3db to agree
C  with their design specs. (STRICTLY EMPIRICAL!!)
C
            RGAIN = 2 * 4 * PI * REW * RNS / ( DNORM * RNORM(ISTA) )
          ELSE
C
C  Set the OOPS gain to GNOOPS.
C
            RGAIN = 10**(GNOOPS/10.)
        ENDIF
C
C  To get the received power from the power density, we multiply by the
C  effective area REFF of the receiving antenna.  This is LAMBDA**2/4*PI
C  for antennas aligned with the E-field.  For isotropically distributed
C  polarization angles, divide by 2.  AMPREC is an intermediate variable
C  used in the calculation of the receiver gain.  It is essentially the
C  received power in milliwatts, WITHOUT the 1/R**2 part, which is added
C  later.  Breaking out the 1/R**2 term avoids arithmetic underflows and
C  log(zero) errors.
C
C
        REFF = WVLN * WVLN / ( 2 * 4 * PI )
        AMPREC = 1000. * REFF * RGAIN * AMPSAT
C
C  Can't let AMPREC be zero.  Otherwise the log blows up.
C
        IF(AMPREC.LT.1.D-20) AMPREC = 1.D-20
C
C  Express the received power in dbm.
C
        AMPL = 10 * ( DLOG10(AMPREC) - DLOG10(4*PI*DRS*DRS) )
        RETURN
        END
```

SUBROUTINE ERROR(ISTA)

```
C
C ***********************************************************************
C *                                                                     *
C *                         SUBROUTINE ERROR                            *
C *                                                                     *
C ***********************************************************************
C
C  AUTHOR:       Dr. E. James Wadiak
C  DATE:         3-FEB-1988
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ.SIM]ERROR.FOR
C
C  CALLING ROUTINE:     SIMDAT.FOR
C
C  SUBROUTINES CALLED:  RANDG.FOR
C
C  COMPILE INSTRUCTIONS:        via INCLUDE statement in SIMDAT.FOR
C
C  LINK/LOAD INSTRUCTIONS:      via INCLUDE statement in SIMDAT.FOR
C
C  PARENT PROGRAM:      SIMDAT.FOR
C
C  PROGRAM DESCRIPTION:
C
C     This subroutine adds normally-distributed random errors to the calculated
C  phase difference values contained in the array DATA.  The RMS magnitude of
C  the errors for each data line are determined by the amplitude of the received
C  signal.  The RMS errors have been modelled by a fourth order polynomial in dB
C  above the assumed noise floor of -152 dBm.  The current error model is based
C  on an analysis of the four hour NAVSPASUR data tape T5321.
C
C    The amplitude and data arrays are passed in the COMMON block.
C
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. Specify the coefficients to the fourth order polynomial fit.
C
C    2. DO  for each data line in the DATA array,
C
C       2a. Calculate the RMS error based on the amplitude of the current data
C           line.
C
C       2b. DO, for each antenna,
C
C           2b(i). Generate an error value having the desired properties.
C
C           2b(ii). IF the datum is not for the reference antenna, add the
C                   phase error to the datum.
C
C           2b(iii). Get the pha-e difference back between -0.5 and +0.5
C                    rotations.
C
C    3. END both DO loops.
C
C    4. Calcculate an amplitude error with the desired properties and add it to
C       the amplitude of the current data line.  (TURN OFF FOR NOW !!)
C
C    5. RETURN to the calling program.
C
C
C  INPUTS   EXPLICIT (via arguments to the CALL statement):
C
C                ISTA - station ID number of the current receiver site.
C
```

```
C           IMPLICIT (via COMMON block):
C
C               DATA(I,L) - calculated (ideal) phase difference for the Ith data
C                           line and the Lth antenna.
C               IAMP(I) - amplitude associated with the Ith data line, in dBm.
C               IX - random number generator seed #1.
C               IY - random number generator seed #2.
C               NANT(ISTA) - number of antennas at receiver ISTA.
C               NLINES - number of data lines in the current data scan.
C
C                    (via assignment statements in algorithm):
C
C               A0 - zeroeth order polynomial fit coefficient.
C               A1 - first order polynomial fit coefficient.
C               A2 - second order polynomial fit coefficient.
C               A3 - third order polynomial fit coefficient.
C               A4 - fourth order polynomial fit coefficient.
C
C  OUTPUTS  EXPLICIT:   NONE
C
C           IMPLICIT (via COMMON block):
C
C               DATA(I,L) - phase difference datum for the Ith data line and
C                           Kth antenna, including random error.
C
C  MAJOR VARIABLES:
C
C               A0 - zeroeth order polynomial fit coefficient.
C               A1 - first order polynomial fit coefficient.
C               A2 - second order polynomial fit coefficient.
C               A3 - third order polynomial fit coefficient.
C               A4 - fourth order polynomial fit coefficient.
C               DATA(I,L) - calculated (ideal) phase difference for the Ith data
C                           line and the Lth antenna.
C               ERVAL - error value returned from subroutine RANDG.
C               IAMP(I) - amplitude associated with the Ith data line, in dBm.
C               ISTA - station ID number of the current receiver site.
C               IX - random number generator seed #1.
C               IY - random number generator seed #2.
C               NANT(ISTA) - number of antennas at receiver ISTA.
C               NDB - amplitude of the current data line, in dB above the noise
C                     floor of -152 dBm.
C               NLINES - number of data lines in the current data scan.
C               RMS - RMS error associated with the amplitude of the current
C                     data line.
C
C
C  MODIFIED:
C
C
C**********************************************************************
        INCLUDE 'SPACE:[WADIAK.LSQ.SIM]SIMDAT.CMN'
C
C  Keep things simple for now.  Calculate an error on the phase only.
C  Use the amplitude-dependent error model generated from the single
C  station least-squares fit to data tape T5321.
C
C  The Gaussian errors are calculated by RANDG with the integers IX
C  and IY controlling the random number generator.
C
        A0 =  1.31D-1
        A1 = -1.23D-2
        A2 =  5.43D-4
        A3 = -4.48D-6
        A4 = -1.02D-7
        DO I=1,NLINES
```

```
      NDB = IDI - IANT(I)
      RMS = A4*NDB**4 + A3*NDB**3 + A2*NDB**2 + A1*NDB + A0
      DO L=1,NANT(ISTA)
        CALL RANDG(IX,IY,RMS,0D0,ERVAL)
        IF(L.NE.NREF(ISTA)) DATA(I,L) = DATA(I,L) + ERVAL
C
C  Get the phase differences back between +/- 0.5 rotations.
C
         IF(DATA(I,L).LT.-0.5) DATA(I,L) = DATA(I,L) + 1.0
         IF(DATA(I,L).GT.+0.5) DATA(I,L) = DATA(I,L) - 1.0
       ENDDO
      ENDDO
C
C  Add random errors of 0.4 dB to the received amplitude.  TURN OFF FOR NOW !!
C
C     CALL RANDG(IX,IY,0.4D0,0D0,VAL)
C     AMPL = AMPL + VAL
      RETURN
      END
      INCLUDE 'SPACE:[WADIAK.LSQ.SIM]RANDG.FOR'
```

```
                         SU     TINI     DG(     ',SI     ?MEA     L)
C
C ***********************************************************************
C *                                                                     *
C *                          SUBROUTINE RANDG                           *
C *                                                                     *
C ***********************************************************************
C
C  AUTHOR:       Dr. E. James Wadiak
C  DATE.         26-JAN-1988
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ.SIM]RANDG.FOR
C
C  CALLING ROUTINES:             ERROR.FOR
C                                MULTLSQ.FOR
C
C  SUBROUTINES CALLED:           RANDU - VAX library subroutine which returns
C                                        a random number uniformly distributed
C                                        between 0 and 1.
C
C  COMPILE INSTRUCTIONS:         via INCLUDE statement in parent program.
C
C  LINK/LOAD INSTRUCTIONS:       via INCLUDE statement in parent program.
C
C  PARENT PROGRAMS:              SIMDAT.FOR, MULTLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C     This subroutine applies the Central Limit Theorem to derivea random
C  number VAL whose distribution is Gaussian with a characteristic dispersion
C  of SIGMA and a mean value of RMEAN.
C
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. Sum 12 random numbers uniformly distributed between 0 and 1.  The
C       resultant number is Gaussian-distributed about the expection value of
C       <6> and a standard deviation pf 1.
C
C    2. Multiply the deviation from the expectation value times the desired
C       standard deviation, and add the desired mean.  This produces a random
C       number with the desired properties.
C
C    3. RETURN to the calling program.
C
C  INPUTS   EXPLICIT (via arguments to the CALL statement):
C
C                IX - random number generator seed.
C                IY - random number generator seed.
C                RMEAN - desired mean of the output random number.
C                SIGMA - desired standard deviation of the output random number.
C
C
C           IMPLICIT:   NONE
C
C  OUTPUTS  EXPLICIT (via the arguments to the CALL statement):
C
C                IX - new seed for next call to RANDG.
C                IY - new seed for next call to RANDG.
C                VAL - random number with the desired distribution properties.
C
C
C           IMPLICIT:   NONE
C
C  MAJOR VARIABLES:
C
```

```
C                 ..  sum .. the .. unifo...y dis....uted ...dom ...bers.
C                 RMEAN - desired mean of output random number.
C                 SIGMA - desired standard deviation of output random number.
C                 VAL - output random number with desired properties.
C
C
C  MODIFIED:
C
C
C*******************************************************************************
        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*4 Y
        A = 0.0
        DO I=1,12
          CALL RANDU(IX,IY,Y)
          A = A + Y
        ENDDO
        VAL = ( A - 6.0 ) * SIGMA + RMEAN
        RETURN
        END
```

```
C
C ***********************************************************************
C *                                                                     *
C *                         PROGRAM MULTLSQ                             *
C *                                                                     *
C ***********************************************************************
C
C  AUTHOR:        Dr. Michael D. Andrews, Dr. E. James Wadiak
C  DATE:          25-FEB-1988
C  LANGUAGE:      FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:          VX7770::SPACE:[WADIAK.LSQ.SIM]MULTLSQ.FOR
C
C  SUBROUTINES CALLED:  DNRMEQ - increments normal equations for Doppler data.
C                       DOP - computes Doppler frequency.
C                       DOPCALC - computes ideal Doppler frequency and chirp.
C                       DOPDERIV - computes Doppler partial derivatives.
C                       MATPR - prints out correlation coefficient matrix.
C                       MCALC - computes ideal phase differences.
C                       MDERIV - computes phase partial derivatives.
C                       NRMEQ - increments normal equations for phase data.
C                       RANDG - computes random Gaussian errors.
C                       RNRMEQ - increments normal equations for chirp data.
C                       SYMIN - solves normal equations by matrix inversion.
C
C  USER-DEFINED         IEL - computes vector subscript for matrix element.
C  FUNCTIONS CALLED:    ERR - computes RMS error based on amplitude.
C
C  COMPILE INSTRUCTIONS:        $ FORTRAN MULTLSQ
C
C  LINK/LOAD INSTRUCTIONS:      $ LINK MULTLSQ
C
C  PROGRAM DESCRIPTION:
C
C     This program will do a nonlinear least squares determination of the
C  satellite position and velocity (and, optionally, station clock offsets) for
C  incoherently combined data of a single satellite pass.  Program input is
C  in the form of calibrated phase differences, Doppler, and chirp, together
C  with predicted satellite position and velocity and time of fence crossing.
C  Each *.DIF input file contains scans for a single satellite with various
C  transimtter/receiver combinations.  All data for a given satellite are
C  processed simultaneously to derive the position and velocity.
C
C     The details of the nonlinear least squares methodology are contained in
C  the report THE DETERMINATION OF NAVSPASUR PHASE DIFFERENCE ERRORS, dated
C  16 March 1988.  Details of the application of the nonlinear least squares
C  technique to simulated NAVSPASUR data using this program are contained in
C  the report NAVSPASUR SYSTEM PERFORMANCE ANALYSIS, dated 29 April 1988.
C
C     The program has been designed to be flexible in the types of data it can
C  handle.  The primary "observed" quantity is differenced phase.  Doppler and
C  chirp data may be included in the solution by setting the proper flags in
C  the MULTLSQ.INP control file.  Other flags and inputs control which of the
C  parameters are solved for, which transmitter/receiver combinations are
C  used in the solution, and what the assumed Doppler and chirp accuracies are.
C
C     The "true" time will be that of the satellite.  In the case of simulated
C  data, no clock offsets need be considered.  For real data, clock offsets
C  may be determined for all the observing stations.  The solution of the
C  satellite position and velocity will be referenced to time TPRED, the
C  predicted time of passage through the fence.  Each data line used in the
C  solution will be tagged with a time TOBS(II), which will be the time offset,
C  in seconds, from time TPRED.
C
C     Almost everything will be passed to and from subroutines via the COMMON
C  block, which also contains most of the array dimensions.  Physical and
```

```
C     yste    nsti    are    .ain    n th    NST.    FOR    ..
C
C
C   PROGRAM ALGORITHM (PSEUDOCODE):
C
C     1. READ in the control flags and inputs which specify the parameters to
C        be solved for, the receiver/transmitter combinations to be used, the
C        assumed Doppler and chirp accuracies, the name of the output file,
C        whether gravitational acceleration is included, and whether the OOPS
C        sites are phase + Doppler or Doppler-only.
C
C     2. READ in the geocentric to great circle rotation angles, the receiver
C        antenna array positions, and the transmitter positions.
C
C     3. Set the control flags NVP and IVP(15) which indicate the number and
C        identity of the parameters which are solved for.
C
C     4. READ the name of the input data file to be processed from a master
C        list of file names.  Return to this point to begin another satellite
C        determination.  On EOF, STOP execution.
C
C     5. READ in the header for the next data scan in the current input file.
C        On EOF, go to step #11.
C
C        6. Introduce random errors on the Doppler and chirp data for simulated
C           data scans.
C
C        7. READ in the geocentric position and velocity of the satellite.
C           Convert these to NAVSPASUR great circle coordinates.
C
C        8. Calculate the Doppler and chirp partial derivatives for the current
C           receiver/transmitter combination.  The partial derivatives are taken
C           with respect to each of the parameters to be varied in the least
C           squares fitting.
C
C        9. READ in the individual phase difference data for all baselines, one
C           timeline at a time.  Store in the DATA array and keep track of the
C           total number of data lines read for the current satellite.
C
C       10. GOTO step #5 to process the next data scan for the current satellite.
C
C    11. ********************************************************************
C        ****************** Begin the least squares fitting. ******************
C        ******** This is the loopback point for iterating the solution. ********
C        ********************************************************************
C        DO, for each data line line that has been read into the data set.
C
C        12. DO, for each phase difference on the current data line.
C
C            13. Calculate the ideal phase difference.
C
C            14. Calculate the residual (observed - calculated) phase difference.
C                Express the result modulo 1 (between -0.5 and +0.5 rotations).
C
C            15. Calculate the phase partial derivatives with respect to each
C                parameter being varied.
C
C            16. Increment the normal equations for the phase datum.
C
C    17. END both DO loops.
C
C    18. IF Doppler data is to be included in the solution THEN,
C
C        19. DO, for each Doppler datum (1 per transmitter/receiver pair),
C
C            20. Calculate tne expected Doppler frequency at time RSEC.
```

```
C
C           21. Calculate the Doppler residual (observed - calculated).
C
C           22. Increment the normal equations for the Doppler datum.
C
C       23. ENDDO
C
C   24. ENDIF
C
C   25. IF chirp data is to be included THEN,
C
C       26. DO, for each chirp datum (1 per receiver/transmitter combination),
C
C           27. Calculate the expected chirp at time RSEC.
C
C           28. Calculate the chirp residual (observed - calculated).
C
C           29. Increment the normal equations for the chirp datum.
C
C       30. ENDDO
C
C   31. ENDIF
C
C   32. Scale the normal equations to minimize roundoff error in inverting.
C
C   33. Solve the normal equations via matrix inversion.
C
C   34. Rescale back to obtain the proper solution.
C
C   35. IF not all parameters meet convergence test and IF not reached the
C       maximum number of iterations, THEN return to step #11 and reiterate
C       the solution.
C
C   36. Once the solution has converged OR reached the maximum number of
C       iterations without converging, calculate some statistical summaries
C       (if converged) and output the results.
C
C   37. GOTO step #4 and begin processing the next satellite pass.
C
C
C  INPUTS (EXPLICIT):
C
C               FOR008 - satellite pass data file containing header information,
C                        predicted satellite position and velocity in geocentric
C                        coordinates, observed Doppler and chirp data, and
C                        differential phase data.  File name is *.DIF.
C
C                        DATA(II,J) - phase difference for the Jth baseline
C                                     at the receiving station and time
C                                     associated with the IIth data line.
C                        DRATE - Doppler rate (chirp) measurement, in Hz/sec.
C                        IALT - satellite altitude, in kilometers.
C                        IAMP(II) - received signal strength for the IIth phase
C                                   difference data line, in units of -dBm.
C                        IDOP - Doppler frequency in Hz.
C                        ISAT - satellite number.
C                        ISTA - receiving station number.
C                        ITRAN - transmitter number.
C                        LONG - satellite longitude, in degrees.
C                        NLINES - number of phase difference data lines.
C                        PSAT(1-3) - predicted satellite geocentric x,y,z
C                                    coordinates at time TPRED.
C                        RSEC - time of the first phase difference data line.
C                        TPRED - predicted time of fence crossing.
C                        VSAT(1-3) - predicted satellite geocentric x,y,z
C                                    velocity components at time TPRED.
C
```

```
C
C           FOR037 - geocentric to great circle rotation angles, in degrees,
C                    and receiving station antenna positions, in meters, in
C                    NAVSPASUR great circle coordinates.
C
C                    ALPHA - geocentric to great circle rotation angle #1,
C                            in degrees west longitude.
C                    BETA - geocentric to great circle rotation angle #2,
C                           in degrees north latitude.
C                    NANT(J) - number of antennas at the Jth receiver site.
C                    NREF(J) - antenna number of the reference antenna at
C                              the Jth receiver site.
C
C                    POS(ISTA,NNT,1):  great circle x,y,z coordinates of the
C                    POS(ISTA,NNT,2):  NNTth antenna at the ISTAth receiver
C                    POS(ISTA,NNT,3):  site, in meters.
C
C
C           FOR038 - positions of the transmitter sites, in meters, in
C                    the NAVSPASUR great circle coordinate system.
C
C                    TPOS(I,1):  NAVSPASUR great circle coordinates of
C                    TPOS(I,2):  the Ith transmitter, in meters.
C                    TPOS(I,3):
C
C
C           FOR040 - control flags and data determining the conditions
C                    of the solution.
C
C                    DELDOP - assumed Doppler accuracy.
C                    DELRAT - assumed chirp accuracy.
C                    FRACTC - convergence test limit.
C                    IGRAV - flag controlling whether gravitational
C                            acceleration is included in solution.
C                    ITMAX - maximum number of iterations allowed.
C                    LAVFIL - name of the error summary file.
C                    LDPLR - flag determining whether OOPS are Doppler-only.
C                    NREC - number of receivers to include in the solution.
C                    NTRAN - number of transmitters to include in solution.
C                    OUTFIL - name of the output file.
C
C
C        (IMPLICIT):  from subroutines via the COMMON block:
C                     ALHS(1-16) - left-hand-sides of the incremented normal
C                                  equations.
C                     CLC - calculated (ideal) phase difference.
C                     DOPRAT - ideal chirp for the current data scan.
C                     RHS(1-16) - right-hand-sides of the incremented normal
C                                 equations.
C
C                     via INCLUDE CONSTANTS.FOR statement:
C                     DTORAD - conversion factor from degrees to radians.
C
C  OUTPUTS (EXPLICIT):
C
C           FOR041 - *.OUT file containing the summary of the least squares
C                    fit, correlation coefficient matrix, position and
C                    velocity errors in each coordinate, in-plane vs.
C                    plane-normal errors, and total errors.
C
C           FOR043 - *.VPT file containing columnar tabulations of the
C                    position and velocity errors as a function of
C                    satellite longitude and altitude.
C
C        (IMPLICIT):  to subroutines via COMMON block:
```

```
C                   DELTA - RMS error associated with the current datum.
C                   DPARTL(ISTA,ITR,J) - partial derivative of the Doppler
C                                        frequency received at station ISTA
C                                        from transmitter ITR, taken with
C                                        respect to the Jth parameter.
C                   PARTL(1-16) - phase difference partial derivatives for
C                                 the current receiver site and antenna,
C                                 taken with respect to each varied
C                                 parameter.
C                   RPARTL(ISTA,ITR,J) - partial derivative of the chirp
C                                        received at station ISTA from
C                                        transmitter ITR, taken with
C                                        respect to the Jth parameter.
C
C  MAJOR VARIABLES:
C
C             ALHS(1-16) - left-hand-sides of the normal equations.
C             ALPHA - geocentric to great circle rotation angle #1,
C                     in degrees west longitude.
C             BETA - geocentric to great circle rotation angle #2,
C                    in degrees north latitude.
C             CA - cos(ALPHA).
C             CB - cos(BETA).
C             DATA(II,J) - phase difference for the Jth baseline
C                          at the receiving station and time
C                          associated with the IIth data line.
C             DELDOP - assumed Doppler accuracy.
C             DELRAT - assumed chirp accuracy.
C             DELTIM - time between the start of the current data scan and
C                      the predicted time of fence crossing.
C             DOPFLG - character string identifying type of OOPS.
C             DOPRAT - ideal (calculated) chirp for the current scan.
C             DPARTL(ISTA,ITR,J) - partial derivative of the Doppler frequency
C                                  frequency received at station ISTA from
C                                  from transmitter ITR, taken with respect
C                                  respect to the Jth parameter.
C             DPLR(I) - Doppler measurement from the Ith data scan.
C             DPRAT(I) - chirp measurement from the Ith data scan.
C             DRATE - Doppler rate (chirp) measurement, in Hz/sec.
C             ERROR(II) - RMS error associated with the IIth data line.
C             FILNAM - name of the output file.
C             FQREC(II) - received frequency for the IIth data line.
C             FRACTC - convergence test limit.
C             GRAV - gravitational constant, in MKS units (0 => no grav).
C             IALT - satellite altitude, in kilometers.
C             IAMP(II) - received signal strength for the IIth phase
C                        difference data line, in units of -dBm.
C             ICNT - counter tracking the total number of data points used in
C                    the solution (phase differences, Dopplers, and chirps).
C             IDOP - Doppler frequency in Hz.
C             IFLAG(1-16) - ID numbers of the parameters being varied.
C             IGRAV - flag controlling whether gravitational
C                     acceleration is included in solution.
C             IRTCNT - number of Doppler/chirp data points.
C             IRX(II) - receiver ID # associated with the IIth data line.
C             ISAT - satellite number.
C             ISTA - receiving station number.
C             ITER - counter keeping track of number of iterations performed.
C             ITMAX - maximum number of iterations allowed.
C             ITRAN - transmitter number.
C             ITX(II) - transmitter ID # associated with the IIth data line.
C             KDOPR(I) - receiver ID # associated with the Ith data scan.
C             KDOPT(I) - transmitter ID # associated with the Ith data scan.
C             LAVFIL - name of the error summary file.
C             LDPLR - flag determining whether OOPS are Doppler-only.
C             LONG - satellite longitude, in degrees.
```

```
C           (J)    mbe     ant     s at     Jth     eiv     .te.
C        NDATA - total number of phase difference data points used.
C        NLINES - number of phase difference data lines.
C        NLNS - total number of data lines for the current satellite.
C        NOOPS - number of OOPS receivers included in the solution.
C        NMPAR(1-16) - names of the possible parameters to vary.
C        NPAR - number of parameter flags to read from FOR040.
C        NPARNC - number of non-converged parameters.
C        NREC - number of receivers to include in the solution.
C        NREF(J) - antenna number of the reference antenna at
C                  the Jth receiver site.
C        NTRAK - number of tracking transmitters included in solution.
C        NTRAN - number of transmitters to include in solution.
C        NVP - number of parameters being varied.
C        OUTFIL - name of the output file.
C        PAR(1-16) - array holding current values for varied parameters.
C        PARTL(1-16) - phase difference partial derivatives for the
C                      current receiver site and antenna, taken with
C                      respect to each varied parameter.
C        PERR - 3-D position error, in meters.
C        PINP - in-plane (2-D) position error, in meters.
C
C        POS(ISTA,NNT,1):  great circle x,y,z coordinates of the
C        POS(ISTA,NNT,2):  NNTth antenna at the ISTAth receiver
C        POS(ISTA,NNT,3):  site, in meters.
C
C        PSAT(1-3) - predicted satellite position at time TPRED.
C        RESID(ICNT) - residual (observed - calculated) for the ICNTth
C                      data point.
C        RHS(1-16) - right-hand-sides of the normal equations.
C        RMAX - maximum phase difference residual.
C        RMEAN - mean of the phase difference residuals.
C        RMIN - minimum phase difference residual.
C        RPARTL(ISTA,ITR,J) - partial derivative of the chirp received at
C                             received at station ISTA from transmitter
C                             transmitter ITR, taken with respect to the
C                             Jth parameter.
C        RRMS - RMS amplitude of the phase difference residuals.
C        RSEC - time of the first phase difference data line in the
C               current scan.
C        RSTRT(I) - start time of the Ith data scan.
C        S(I) - scale factor for the Ith normal equation.
C        SA - sin(ALPHA).
C        SB - sin(BETA).
C        SIGM(I) - formal error on the Ith parameter.
C
C        SIGXKM:
C        SIGYKM: x,y,z velocity errors, in kilometers/minute.
C        SIGZKM:
C
C        TOBS(II) - time of the IIth data line (with respect to TPRED).
C
C        TPOS(I,1):  NAVSPASUR great circle coordinates of
C        TPOS(I,2):  the Ith transmitter, in meters.
C        TPOS(I,3):
C
C        TPRED - predicted time of fence crossing.
C        VERR - 3-D velocity error, in meters/second.
C        VERRK - 3-D velocity error, in kilometers/minute.
C        VINP - in-plane (2-D) velocity error, in kilometers/minute.
C        VSAT(1-3) - predicted satellite velocity at time TPRED.
C
C
C  FORMAT STATEMENT LABELS USED:
C           401,402,403,404,405,406,407,408,409,410
C
```

```
C  MODIFIED:
C
C
C*****************************************************************************
        INCLUDE 'SPACE:[WADIAK.LSQ.SIM]MULTLSQ.CMN/LIST'
        CHARACTER*14 DOPFLG
        CHARACTER*42 FILNAM,OUTFIL,LAVFIL
        DATA NMPAR/'X','Y','Z','VX','VY','VZ','CL1','CL2',
     +             'CL3','CL4','CL5','CL6','CL7','CL8','CL9','CL10'/
        INCLUDE 'SPACE:[WADIAK.LSQ.SIM]CONSTANTS.FOR/LIST'
C
C  Read in the number of receivers and transmitters in the data set.
C
        READ(40,*) NREC,NTRAN
C
C  Read in 6+NREC flags (0 or 1) to control which parameters are solved
C  for.  The first 6 flags represent satellite position and velocity,
C  respectively.  The remaining flags refer to the receiving station
C  clock offsets.  Set the clock offsets equal to zero for now.
C
        NPAR = 6 + NREC
        READ(40,*) (IFLAG(I),I=1,NPAR)
        DO I=7,NPAR
          PAR(I) = 0.0
        ENDDO
C
C  Read in the flag controlling whether the acceleration term is
C  included in the calculations.
C
        READ(40,*) IGRAV
        IF(IGRAV.EQ.0) GRAV = 0.
C
C  Read in the RMS accuracies of the Doppler frequency (in Hz) and
C  the Doppler rate (in Hz/sec).  ( 0 => not used in solution)
C
        READ(40,*) DELDOP
        READ(40,*) DELRAT
C
C  Read in the flag controlling whether the OOPS are Doppler-only.
C  If LDPLR = 1, then the OOPS are Doppler-only.
C
        READ(40,*) LDPLR
        DOPFLG = ' PHASE+Doppler'
        IF(LDPLR.EQ.1) DOPFLG = ' Doppler ONLY '
C
C  Read in the maximum number of iterations and the convergence test
C  limit.  Iteration ends after ITMAX iterations, or after all
C  corrections are less than FRACTC times the error.
C
        READ(40,*) ITMAX,FRACTC
C
C  Read in the name of the output file and open it.
C
        READ(40,*) OUTFIL
        OPEN(UNIT=41,FILE=OUTFIL,STATUS='NEW')
        READ(40,*) LAVFIL
        OPEN(UNIT=43,FILE=LAVFIL,STATUS='NEW')
        CLOSE(40)
C
C  The rotation angles for the transformation to great circle
C  coordinates are the geocentric latitude and longitude corresponding
C  to the geocentric X,Y,Z coordinates of the fence as given in the
C  NAVSPASUR coordinates list ( array B(3,10,101) in COMMON area COOR ).
C  Input the geocentric to great circle rotation angles from RECGC.POS.
C
```

```
      READ(37,*) ALPHA,BETA
      ALPHA = ALPHA * DTORAD
      BETA  = BETA  * DTORAD
      SA = DSIN(ALPHA)
      CA = DCOS(ALPHA)
      SB = DSIN(BETA)
      CB = DCOS(BETA)
C
C  Input the number of antennas at each of the receiver sites and
C  the antenna number of the reference antenna at each site.
C
      READ(37,*) (NANT(J),J=1,NREC)
      READ(37,*) (NREF(J),J=1,NREC)
C
C  Input antenna and station positions from RECGC.POS for NREC
C  stations (presently, COMMON statement limits NREC to 10 or less).
C  The positions in the input file are currently in great circle
C  coordinates.
C
      DO ISTA=1,NREC
        DO NNT=1,NANT(ISTA)
          READ(37,*) (POS(ISTA,NNT,J),J=1,3)
        ENDDO
C
C  Decrement NANT(ISTA) to reflect the number of phase DIFFERENCES at
C  each site.
C
        NANT(ISTA) = NANT(ISTA) - 1
      ENDDO
      CLOSE(37)
C
C  Input the transmitter positions from TRANGC.POS.
C
      READ(38,*)
      DO I=1,NTRAN
        READ(38,*) (TPOS(I,J),J=1,3)
      ENDDO
      CLOSE(38)
C
C  Do some bookeeping stuff so we can keep track of things.
C
      NVP=0
      DO I=1,NPAR
        NVP = NVP + IFLAG(I)
      ENDDO
      DO I=1,NVP
        IVP(I) = 0
      ENDDO
      JJJ = 0
      DO I=1,NPAR
        IF(IFLAG(I).EQ.1) JJJ = JJJ + 1
        IF(IFLAG(I).EQ.1) IVP(JJJ) = I
      ENDDO
C
C  Get the name of the data file to be processed and open for input.
C
C  THIS IS THE LOOPBACK POINT FOR BEGINNING EACH NEW DATA SET.
C
 300  READ(42,406,END=888) FILNAM
 406  FORMAT(A42)
      WRITE(6,407) FILNAM
 407  FORMAT(2X,A42)
      OPEN(UNIT=8,FILE=FILNAM,STATUS='OLD')
C
C  Read in the data.  For each scan there will be NLINES data lines.
C  The counter II will keep a running total of the total number of data
```

```
C  ..nes ...d in ... sol....n, ..... IR.... cou... the ...ber o.
C  receiver/transmitter combinations (used to index the Doppler data).
C  Read the header line.  This is the loopback point to read all the
C  data scans in the data set.  The header line contains the transmitter
C  index, the receiver index, the number of antennas, the number of scans,
C  the transmitter frequency, the measured Doppler, and the start time of
C  the data.
C
       II = 0
       IRTCNT = 0
 301   READ(8,*,END=302,ERR=999)ISAT,ISTA,ITRAN,IDOP,RSEC,TPRED,NLINES,
     +      DRATE,IALT,LONG
C
C  Bypass the scan if ISTA > NREC or ITRAN - 6 > NTRAN.
C
       ITRAN = ITRAN - 6
       IF(ISTA.GT.NREC.OR.ITRAN.GT.NTRAN) THEN
         DO ISKIP=1,NLINES+1
           READ(8,*)
         ENDDO
         GOTO 301
       ENDIF
       IRTCNT = IRTCNT + 1
C
C  In the data generation program, no errors are introduced on the
C  measured Doppler frequency or rate.  This allows us to explore
C  the effect of different errors using the same input data set.  Add
C  Doppler errors here.  The errors are assumed to be Gaussian with
C  means of zero and dispersions of DELDOP and DELRAT.
C
       CALL RANDG(IX,IY,DELDOP,0.0,DOPERR)
       DPLR(IRTCNT) = FLOAT(IDOP) + DOPERR
       CALL RANDG(IX,IY,DELRAT,0.0,RATERR)
       DPRAT(IRTCNT) = DRATE + RATERR
       RSTRT(IRTCNT) = RSEC
C
C  We need to have some way to keep the books on which RX/TX pair is
C  associated with each Doppler datum.  The arrays KDOPR and KDOPT will
C  do this.
C
       KDOPR(IRTCNT) = ISTA
       KDOPT(IRTCNT) = ITRAN
C
C  Read in the predicted satellite position and velocity (in geocentric
C  coordinates) at time TPRED.  Convert to great circle coordinates.
C  Set the initial values for the parameters based on the input position
C  at time TPRED.
C
       READ(8,*) PSAT,VSAT
       PX = PSAT(1)
       PY = PSAT(2)
       PZ = PSAT(3)
       VX = VSAT(1)
       VY = VSAT(2)
       VZ = VSAT(3)
       PSAT(1) = (CB*CA)*PX + (CB*SA)*PY - (SB)*PZ
       VSAT(1) = (CB*CA)*VX + (CB*SA)*VY - (SB)*VZ
       PSAT(2) =  - (SA)*PX +     (CA)*PY
       VSAT(2) =  - (SA)*VX +     (CA)*VY
       PSAT(3) = (CA*SB)*PX + (SA*SB)*PY + (CB)*PZ
       VSAT(3) = (CA*SB)*VX + (SA*SB)*VY + (CB)*VZ
       DO I=1,3
         PAR(I) = PSAT(I)
         PAR(I+3) = VSAT(I)
       ENDDO
C
```

```
C  Check whether the station is an OOPS  AND** .. the
C  OOPS stations are Doppler-only.  If so, skip over the
C  phase data.
C
       IF(ISTA.GT.6.AND.LDPLR.EQ.1) THEN
           DO I=1,NLINES
             READ(8,*)
           ENDDO
         ELSE
           DO I=1,NLINES
             II = II + 1
             READ(8,*) IAMP(II),(DATA(II,J),J=1,NANT(ISTA))
             ERROR(II) = ERR(IAMP(II))
             TOBS(II) = RSEC + (I - 0.5) * TINCR - TPRED
C
C  The following uses the NAVSPASUR Doppler convention
C  where + => BLUESHIFT.
C
             FQREC(II) = FREQ + DPLR(IRTCNT)
             IRX(II) = ISTA
             ITX(II) = ITRAN
           ENDDO
       ENDIF
C
C  Calculate the Doppler and Doppler rate partial derivatives.
C  The time of the calculations will be RSEC.  The Doppler
C  subroutines use time offsets from TPRED.  The calculated
C  values are returned in the common array.
C
       DELTIM = RSEC - TPRED
       CALL DOPDERIV(ISTA,ITRAN,DELTIM)
       GOTO 301
C
C  Come here when all the data has been read in.  If there isn't any
C  data, bail out!!
C
302    NLNS = II
       IF(NLNS.EQ.0) GOTO 999
C
C  If there are data points, write some header info to the output file.
C
       NOOPS = NREC - 6
       NTRAK = NTRAN - 3
       WRITE(41,405) LONG,IALT,NOOPS,DOPFLG,NTRAK,DELDOP,DELRAT
 405   FORMAT('1 Longitude ',I3,' degrees    Altitude ',I5,' km',//,
     +          '  Solution for ',I1,A14,' OOPS and ',I1,' TRACKERS',//,
     +          '  Doppler accuracy in Hz: ',F5.1,/,
     +          '    (0 => NO Doppler)',//,
     +          '  Doppler rate accuracy in Hz/sec: ',F5.1,/,
     +          '    (0 => NO Doppler RATE)',/)

C
C  Start the Least Squares fitting.
C
C  ******************************************************************
C  ****   This is the loopback point for iterating the solution.   ****
C  ******************************************************************
C
       ITER = 0
303    ITER = ITER + 1
C
C  This is an iterative process so it is NECESSARY to initialize some
C  arrays on each pass.
C
       K = IEL(NVP,NVP)
       DO I=1,K
```

```
         ..(I.L....P+1) ...N
           RHS(I) = 0.
           PARTL(I) = 0.
         ENDIF
         ALHS(I) = 0.
       ENDDO
C
C  Loop through the scans and the antennas within a scan.
C  The work is all done in the subroutines.  The subroutines MCALC
C  and MDERIV are problem specific i.e., the nature of the observables
C  must be considered.  The remaining subroutines are general.  They
C  are used to set-up and invert the matrices and display the results.
C
       ICNT = 0
       DO II = 1,NLNS
         LLL = IRX(II)
         DO I=1,NANT(LLL)
         ICNT = ICNT + 1
C
C  Call MCALC to calculate the "ideal" value of the phase difference.
C  Adjust the phase difference to lie between -0.5 and + 0.5.
C
           CALL MCALC(II,I)
           VRBL = DATA(II,I) - CLC
           DO WHILE (VRBL.LT.-0.5)
             VRBL = VRBL + 1.
           ENDDO
           DO WHILE (VRBL.GT.+0.5)
             VRBL = VRBL - 1.
           ENDDO
           RESID(ICNT) = VRBL
C
C  Get the phase partial derivatives and increment the normal equations.
C
           CALL MDERIV(II,I)
           DELTA = ERROR(II)
           CALL NRMEQ(ICNT)
         ENDDO
       ENDDO
C
C  NDATA gives the total number of phase data points.  It is needed
C  later on when the residual statistics are done.
C
       NDATA = ICNT
C
C  Include the Doppler data in the least-squares fit here.  There is
C  one Doppler datum for each receiver/transmitter site (IRTCNT sites
C  in all).  Calculate the "ideal" Doppler shift for the current
C  parameter values, compute the Doppler residuals, and increment
C  the normal equations appropriately.
C
C  If DELDOP equals zero, no Doppler is to be included in the solution.
C
       IF(DELDOP.GT.0.0) THEN
         DELTA = DELDOP
         DO KD=1,IRTCNT
           DELTIM = RSTRT(KD) - TPRED
           CALL DOP(KDOPR(KD),KDOPT(KD),DELTIM)
           ICNT = ICNT + 1
           RESID(ICNT) = DPLR(KD) - DOPLR
           CALL DNRMEQ(KD,ICNT)
         ENDDO
       ENDIF
C
C  Same as above for Doppler rate.
C
```

```
            ELR/     .0.     IEN
          DELTA = DELRAT
          DO KD=1,IRTCNT
            DELTIM = RSTRT(KD) - TPRED
            CALL DOPCALC(KDOPR(KD),KDOPT(KD),DELTIM)
            ICNT = ICNT + 1
            RESID(ICNT) = DPRAT(KD) - DOPRAT
            CALL RNRMEQ(KD,ICNT)
          ENDDO
        ENDIF
C  Get the square roots of the diagonal elements and use those
C  values to scale the array which will insure numerical stability.
C
        DO I=1,NVP
          JJ = IEL(I,I)
          S(I) = DSQRT( ALHS(JJ) )
          IF(S(I).EQ.0D0) S(I) = 1.
        ENDDO
        DO I=1,NVP
          RHS(I) = RHS(I)/S(I)
          DO J=1,I
            K = IEL(I,J)
            ALHS(K) = ALHS(K) / ( S(I) * S(J) )
          ENDDO
        ENDDO
C
C  Solve the normal equations in subroutine SYMIN.
C
        CALL SYMIN
C
C  Rescale the result to get back to the correct units.
C
        DO I=1,NVP
          RHS(I) = RHS(I) / S(I)
          DO J=1,I
            K = IEL(I,J)
            ALHS(K) = ALHS(K) / ( S(I) * S(J) )
          ENDDO
        ENDDO
C
C  Calculate the correction matrix.
C
        DO I=1,NVP
          J = IEL(I,I)
          SIGM(I) = DSQRT(.ALHS(J) )
        ENDDO
        DO I=1,NVP
          DO J=1,I
            IA = IEL(I,I)
            IB = IEL(J,J)
            A = DSQRT( ALHS(IA) )
            B = DSQRT( ALHS(IB) )
            K = IEL(I,J)
            CORR(K) = ALHS(K) / ( A * B )
          ENDDO
        ENDDO
C
C  ANALYZE THE RESIDUALS BY FINDING THE LARGEST, SMALLEST, MEAN,
C  AND THE RMS.
C
        RMAX = -1.E30
        RMIN = 1.E30
        RMEAN = 0.
        RRMS = 0.
        DO I=1,NDATA
          AR = RESID(I)
```

```
          (AR.    MAX    .X =
          IF(AR.LT.RM N) RMIN = AR
          RMEAN = RMEAN + AR
        ENDDO
        RMEAN = RMEAN / NDATA
        DO I=1,NDATA
          AR = RESID(I)
          RRMS=RRMS + AR * AR
        ENDDO
        RRMS = DSQRT( RRMS / NDATA )
C
C  Calculate updated parameters.
C
        NPARNC = 0
        JFLG = 0
        DO I=1,NVP
          J = IVP(I)
          A = PAR(J) + RHS(I)
          PAR(J) = A
          FRACT = RHS(I) / SIGM(I)
          FRACT = DABS( FRACT )
          IF(FRACT.GE.FRACTC) THEN
            JFLG = 1
            NPARNC = NPARNC + 1
          ENDIF
        ENDDO
        IF(JFLG.EQ.0.OR.ITER.EQ.ITMAX) GOTO 304
        GOTO 303
304     WRITE(41,409) JFLG,FRACTC,ITER,ITMAX,NPARNC
409     FORMAT(//,' Iteration terminated with JFLG = ',I1,
     +        ' FRACTC = ',0PF4.2,' ITER = ',I3,' ITMAX = ',
     +        I3,/,1X,I3,' parameters not converged',/)
C
C  Print out the initial and final parameter values.
C
        WRITE(41,401) PSAT,VSAT,(PAR(J),J=1,NVP)
 401    FORMAT(/,' Initial, final values of each parameter:',/,
     +         2X,3F14.2,3F10.2/,2X,3F14.2,3F10.2)
C
C  Write out the error estimates for each parameter.
C
        SIGXKM = SIGM(4) * 60. / 1000.
        SIGYKM = SIGM(5) * 60. / 1000.
        SIGZKM = SIGM(6) * 60. / 1000.
        WRITE(41,404) (SIGM(I),I=1,6),SIGXKM,SIGYKM,SIGZKM
 404    FORMAT (/,' Uncertainties in each parameter:',/,
     +     8X,'dX',8X,'dY',8X,'dZ',/,2X,3F10.2,4X,'meters',/,
     +     7X,'dVX',7X,'dVY',7X,'dVZ',/,2X,3F10.2,4X,'meters/sec',/,
     +     2X,3F10.2,4X,'km/min',/)
C
C  Calculate write out the TOTAL position and velocity errors.
C
        PERR = 0.0
        VERR = 0.0
        DO J=1,3
          DO K=1,J
            IF(J.EQ.K) THEN
                RNUM = 1.
              ELSE
                RNUM = 2.
            ENDIF
            JJ = IEL(J,K)
            KK = IEL(J+3,K+3)
            PERR = PERR + DABS( RNUM*CORR(JJ)*SIGM(J)*SIGM(K) )
            VERR = VERR + DABS( RNUM*CORR(KK)*SIGM(J+3)*SIGM(K+3) )
          ENDDO
```

```
C  Calculate in-plane errors separately.
C
            IF(J.EQ.2) THEN
              PINP = DSQRT( PERR )
              VINP = DSQRT( VERR ) * 60. / 1.D3
            ENDIF
          ENDDO
          PERR = DSQRT( PERR )
          VERR = DSQRT( VERR )
          VERRK = VERR * 60. / 1.D3
          PERRKM = PERR / 1.D3
          WRITE(41,402) PINP,PERR,VINP,VERRK
 402      FORMAT(/,'  THE UNCERTAINTIES IN POSITION AND VELOCITY ARE:',/,
     +    '  delta R = ',F9.2,' meters in-plane, ',F9.2,' meters total',/,
     +    '  delta V = ',F9.2,' km/min in-plane, ',F9.2,' km/min total',/)
C
C  Write out the longitude, altitude, velocity and position errors to
C  the .VPT file.  Avoid output conversion errors by setting excessive
C  values to -999.
C
          IF(VERRK.GE.999.) VERRK = -999.
          IF(VINP.GE.999.) VINP = -999.
          IF(SIGZKM.GE.999.) SIGZKM = -999.
          IF(PERRKM.GE.999.) PERRKM = -999.
          WRITE(43,408) LONG,IALT,VERRK,VINP,SIGZKM,PERRKM
 408      FORMAT(2X,I3,I7,3F9.2,8X,F9.2)
C
C  Write out the covariance matrix if convergence is achieved.
C
          IF(JFLG.GT.0) GOTO 999
          WRITE(41,410)
 410      FORMAT(/,' THE CORRELATION MATRIX IS',/)
          CALL MATPR(CORR,1,NVP,41)
C
C  Analyze and print out information on the final residuals.
C
          RMAX = -1.E30
          RMIN = 1.E30
          RMEAN = 0.
          RRMS = 0.
          DO I=1,NDATA
            AR = RESID(I)
            IF(AR.GT.RMAX) RMAX = AR
            IF(AR.LT.RMIN) RMIN = AR
            RMEAN = RMEAN + AR
          ENDDO
          RMEAN = RMEAN / NDATA
          DO I=1,NDATA
            AR = RESID(I)
            RRMS = RRMS + AR*AR
          ENDDO
          RRMS = DSQRT( RRMS / NDATA )
          WRITE(41,403) RMAX,RMIN,RMEAN,RRMS
 403      FORMAT(/,' Distribution of final residuals:',/,' MAX = ',
     +      F7.4,' MIN = ',F7.4,/,' MEAN = ',F7.4,' RMS = ',F7.4,/)
 999      GOTO 300
 888      STOP
          END
          INCLUDE 'SPACE:[WADIAK.LSQ.SIM]MCALC.FOR/LIST'
          INCLUDE 'SPACE:[WADIAK.LSQ.SIM]MDERIV.FOR/LIST'
          INCLUDE 'SPACE:[WADIAK.LSQ.SIM]ERR.FOR/LIST'
          INCLUDE 'SPACE:[WADIAK.LSQ.SIM]DOP.FOR/LIST'
          INCLUDE 'SPACE:[WADIAK.LSQ.SIM]DOPCALC.FOR/LIST'
          INCLUDE 'SPACE:[WADIAK.LSQ.SIM]DOPDERIV.FOR/LIST'
          INCLUDE 'SPACE:[WADIAK.LSQ.SIM]RANDG.FOR/LIST'
```

```
...UDE  '...CE:[...AK....SIM]....FOR/....'
INCLUDE 'SPACE:[WADIAK.LSQ.SIM]NRMEQ.FOR/LIST'
INCLUDE 'SPACE:[WADIAK.LSQ.SIM]DNRMEQ.FOR/LIST'
INCLUDE 'SPACE:[WADIAK.LSQ.SIM]RNRMEQ.FOR/LIST'
INCLUDE 'SPACE:[WADIAK.LSQ.SIM]SYMIN.FOR/LIST'
INCLUDE 'SPACE:[WADIAK.LSQ.SIM]MATPR.FOR/LIST'
```

```
C****************************************************************************
C*                                                                          *
C*                      COMMON BLOCK MULTLSQ                                *
C*                                                                          *
C****************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON   IRTCNT,LU,NDATA,NREC,NVP,NANT(10),NREF(10),NMPAR(16),
     +  IANS(16),IFLAG(16),IVP(16),KDOPR(50),KDOPT(50),
     +  IAMP(2725),IRX(2725),ITX(2725),
     +  AMPL,CLC,DELTA,DOPLR,DOPRAT,DRATE,
     +  FREQ,GRAV,SIGMA,TPRED,VLIGHT,
     +  ASAT(3),PSAT(3),VSAT(3),PHASE(12),PAR(16),PARTL(16),RHS(16),
     +  S(16),SIGM(16),DPLR(50),DPRAT(50),
     +  RSTRT(50),ALHS(136),CORR(136),
     +  ERROR(2725),TOBS(2725),FQREC(2725),RESID(32700),TPOS(5,3),
     +  DATA(2725,11),DPARTL(10,5,16),RPARTL(10,5,16),POS(10,12,3)
```

```
C
C ***********************************************************************
C *                                                                     *
C *                     SUBROUTINE DNRMEQ                               *
C *                                                                     *
C ***********************************************************************
C
C AUTHOR:       Dr. E. James Wadiak
C DATE:         25-FEB-1988
C LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C FILE:         VX7770::SPACE:[WADIAK.LSQ.SIM]DNRMEQ.FOR
C
C CALLING ROUTINE:      MULTLSQ.FOR
C
C SUBROUTINES CALLED:   NONE
C
C USER-DEFINED          IEL.FOR
C FUNCTIONS CALLED:
C
C COMPILE INSTRUCTIONS:         via INCLUDE statement in MULTLSQ.FOR
C
C LINK/LOAD INSTRUCTIONS:       via INCLUDE statement in MULTLSQ.FOR
C
C PARENT PROGRAM:       MULTLSQ.FOR
C
C PROGRAM DESCRIPTION:
C
C    This subroutine increments the normal equations of the nonlinear least
C squares fitting program MULTLSQ.  Each call to DNRMEQ increments the normal
C equations for one Doppler datum residual.
C
C PROGRAM ALGORITHM (PSEUDOCODE):
C
C   1. Get the error associated with the datum via the COMMON block.  Set the
C      data weight equal to 1./(error)**2.
C
C   2. DO, for each varied parameter (i.e., NVP normal equations).
C
C      2a. Increment the right-hand-side of the Ith normal equation by the
C          weighted data residual times the partial derivative of the Doppler
C          taken with respect to the Ith parameter.
C
C      2b. DO, for each lower triangular matrix element on the left-hand-side
C          of the Ith normal equation,
C
C          2b(1). Increment the left-hand-side by the weighted product of the
C                 partial derivatives of the Doppler frequency taken with
C                 respect to the Ith (row) and Jth (column) parameters.
C
C   3. END both DO loops.
C
C   4. RETURN to the calling program.
C
C INPUTS    EXPLICIT (via arguments to the CALL statement):
C
C               ID - array index specifying which Doppler datum to use.
C               IOBS - array index specifying which residual datum to use.
C
C           IMPLICIT (via COMMON block):
C
C               ALHS(1-16) - current left-hand-sides of the normal equations.
C               DELTA - RMS error associated with the IOBSth data residual.
C               DPARTL(IR,IT,I) - partial derivative of the Doppler frequency
C                                 of the signal from transmitter IT received
C                                 at station IR, taken with respect to the
```

```
                                  :h p    ster
C             IVP(1-6) - array containing the parameter ID numbers of each of
C                        the NVP parameters being varied.
C             NVP - number of parameters being varied in least squares fit.
C             RHS(1-16) - current right-hand-sides of the normal equations.
C
C
C  OUTPUTS  EXPLICIT:   NONE
C
C           IMPLICIT (via COMMON block):
C
C             ALHS(1-16) - incremented left-hand-sides of normal equations.
C             RHS(1-16) - incremented right-hand-sides of normal equations.
C
C
C  MAJOR VARIABLES:
C
C             ALHS(1-16) - left-hand-sides of the normal equations.
C             DELTA - RMS error associated with the IOBSth residual.
C             DPARTL(IR,IT,I) - partial derivative of the Doppler frequency
C                               of the signal from transmitter IT received
C                               at station IR, taken with respect to the
C                               Ith parameter.
C             GG - weight assigned to the current datum.
C
C
C  MODIFIED:
C
C
C*****************************************************************************
        INCLUDE 'SPACE:[WADIAK.LSQ.SIM]MULTLSQ.CMN/LIST'
      GG = 1. / ( DELTA * DELTA )
      DO I=1,NVP
        IA = IVP(I)
        PARTLA = DPARTL(KDOPR(ID),KDOPT(ID),IA)
        RHS(I) = RHS(I) + GG * RESID(IOBS) * PARTLA
        DO J=1,I
          K = IEL(I,J)
          IB = IVP(J)
          PARTLB = DPARTL(KDOPR(ID),KDOPT(ID),IB)
          ALHS(K) = ALHS(K) + GG * PARTLA * PARTLB
        ENDDO
      ENDDO
      RETURN
      END
```

```
                    OUT:     OP(      ,ITR     (ME)
C
C ****************************************************************************
C *                                                                          *
C *                            SUBROUTINE DOP                                *
C *                                                                          *
C ****************************************************************************
C
C  AUTHOR:        Dr. E. James Wadiak
C  DATE:          3-FEB-1988
C  LANGUAGE:      FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:          VX7770::SPACE:[WADIAK.LSQ.SIM]DOP.FOR
C
C  CALLING ROUTINES:     MULTLSQ.FOR
C                        DOPCALC.FOR
C
C  SUBROUTINES CALLED:  NONE
C
C  COMPILE INSTRUCTIONS:         via INCLUDE statement in MULTLSQ.FOR.
C
C  LINK/LOAD INSTRUCTIONS:       via INCLUDE statement in MULTLSQ.FOR.
C
C  PARENT PROGRAM:      MULTLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C     This subroutine calculates the expected Doppler shift frequency at time
C  TIME for the radar carrier wave transmitted from transmitter site ITRAN and
C  received at receiver site ISTA.  The position and velocity of the satellite
C  at time TIME, as well as the transmitter and receiver site coordinates, are
C  passed to the subroutine in the COMMON block.  The result is returned to the
C  calling program via the COMMON block.
C
C     The coordinate system is the NAVSPASUR great circle coordinate system used
C  throughout the MULTLSQ program suite.
C
C     The Doppler sign convention is the NAVSPASUR convention of + => BLUESHIFT.
C
C     The acceleration term due to gravity is included in the calculations.  It
C  can be (and ususally is) turned off via a flag in the main program which sets
C  the gravitational constant GRAV equal to zero.
C
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. Calculate the acceleration at time TPRED (the predicted time of fence
C       crossing), using a Keplerian gravitational potential.  If the accel-
C       eration flag has been turned off in the main program, the value of the
C       gravitational constant GRAV, which is passed via the COMMON block, is
C       zero, so that the acceleration is zero.  Calculate the x,y,z components
C       of the acceleration vector in NAVSPASUR great circle coordinates.
C
C    2. Calculate the satellite position at time TIME via X + VT +AT**2/2.
C
C    3. Calculate the satellite velocity at time TIME.
C
C    4. Calculate the transmitter-satellite and receiver-satellite position
C       vectors.
C
C    5. Calculate the dot product of the transmitter-satellite unit position
C       vector and the satellite velocity vector.
C
C    6. Calculate the dot product of the receiver-satellite unit position vector
C       and the satellite velocity vector.
C
C    7. Calculate the Doppler-shifted radar frequency at the satellite.
```

```
C    8. Calculate the (twice) Doppler-shifted radar frequency at the receiver.
C
C    9. Calculate the Doppler shift as the difference between the transmit
C       frequency and the receive frequency.
C
C   10. RETURN to the calling program.
C
C
C  INPUTS   EXPLICIT (via arguments to the CALL statement):
C
C               ISTA - receiver station ID number.
C               ITRAN - transmitter station ID number.
C               TIME - time of the calculation relative to TPRED, in seconds.
C
C           IMPLICIT (via COMMON block):
C
C               EARTHM - mass of the earth, in kilograms.
C               FREQ - NAVSPASUR transmit frequency, in Hz.
C               GRAV - universal gravitational constant, in MKS units (set to
C                      zero when acceleration turned off).
C               NREF(ISTA) - antenna ID number of the reference antenna at
C                            receiver site ISTA.
C               PAR(1-6) - current estimates of the satellite position and
C                          velocity components at time TPRED (time of fence
C                          crossing), in meters and meters/second.
C               POS(ISTA,NREF,1-3) - x,y,z position of the reference antenna at
C                                    receiver station ISTA.
C               TPOS(ITRAN,1-3) - x,y,z position of transmitter ITRAN.
C               VLIGHT - velocity of light, in meters/second.
C
C
C  OUTPUTS  EXPLICIT:   NONE
C
C           IMPLICIT (via the COMMON block):
C
C               DOPLR - calculated Doppler shift.
C
C  MAJOR VARIABLES:
C
C               ACCEL - magnitude of the satellite acceleration at time TPRED,
C                       in meters/second**2.
C               ASAT(1-3) - satellite acceleration x,y,z components at time
C                           TPRED, in meters/second**2.
C               DOPLR - calculated Doppler shift.
C               DS0 - geocentric distance to satellite at time TPRED, in meters.
C               DTS - transmitter-satellite distance, in meters.
C               EARTHM - mass of the earth, in kilograms.
C               FREQ - NAVSPASUR transmit frequency, in Hz.
C               FREQ1 - radar frequency at the satellite, in Hz.
C               FREQ2 - radar frequency at the receiver, in Hz.
C               GRAV - universal gravitational constant, in MKS units (set to
C                      zero when acceleration turned off).
C               ISTA - receiver station ID number.
C               ITRAN - transmitter station ID number.
C               NREF(ISTA) - antenna ID number of the reference antenna at
C                            receiver site ISTA.
C               PAR(1-6) - current estimates of the satellite position and
C                          velocity components at time TPRED (time of fence
C                          crossing), in meters and meters/second.
C               POS(ISTA,NREF,1-3) - x,y,z position of the reference antenna at
C                                    receiver station ISTA.
C               RSDOTV - dot product of the receiver-satellite unit position
C                        vector and satellite velocity vector, in meters/second.
C
C               RSX:
```

```
C               RSY: receiver-satellite position vector components at time TIME,
C               RSZ: in meters.
C
C               RTS - receiver-satellite distance, in meters.
C               TIME - time of the calculation relative to TPRED, in seconds.
C               TPOS(ITRAN,1-3) - x,y,z position of transmitter ITRAN.
C               TSDOTV - dot product of the transmitter-satellite unit position
C                        vector and satellite velocity vector, in meters/second.
C
C               TSX:
C               TSY: transmitter-satellite position vector components at TIME,
C               TSZ: in meters.
C
C               VSX:
C               VSY: satellite velocity components at TIME, in meters/second.
C               VSZ:
C
C               X:
C               Y: satellite position components at TIME, in meters.
C               Z:
C
C
C  MODIFIED:
C
C
C*****************************************************************************
       INCLUDE 'SPACE:[WADIAK.LSQ.SIM]MULTLSQ.CMN/LIST'
C
C  ASAT gives the satellite's acceleration components at time TPRED.  The
C  acceleration term can be turned off via a flag in the main program, which
C  makes the gravitational constant GRAV equal to zero.
C
       DS0 = DSQRT(PAR(1)*PAR(1) + PAR(2)*PAR(2) + PAR(3)*PAR(3))
       ACCEL = - GRAV * EARTHM / ( DS0 * DS0 )
       ASAT(1) = ACCEL * PAR(1) / DS0
       ASAT(2) = ACCEL * PAR(2) / DS0
       ASAT(3) = ACCEL * PAR(3) / DS0
C
C  X,Y,Z give the satellite position at time TIME.
C
       X = PAR(1) + PAR(4) * TIME + ASAT(1) * TIME * TIME / 2.
       Y = PAR(2) + PAR(5) * TIME + ASAT(2) * TIME * TIME / 2.
       Z = PAR(3) + PAR(6) * TIME + ASAT(3) * TIME * TIME / 2.
       DS = DSQRT( X*X + Y*Y + Z*Z )
C
C  Calculate the satellite velocity at time TIME.
C
       VSX = PAR(4) + ASAT(1) * TIME
       VSY = PAR(5) + ASAT(2) * TIME
       VSZ = PAR(6) + ASAT(3) + TIME
C
C  TSX,TSY,TSZ give the satellite position relative to the transmitter.
C
       TSX = X - TPOS(ITRAN,1)
       TSY = Y - TPOS(ITRAN,2)
       TSZ = Z - TPOS(ITRAN,3)
       DTS = DSQRT( TSX*TSX + TSY*TSY + TSZ*TSZ )
C
C  Repeat the above for the receiver.
C
       RSX = X - POS(ISTA,NREF(ISTA),1)
       RSY = Y - POS(ISTA,NREF(ISTA),2)
       RSZ = Z - POS(ISTA,NREF(ISTA),3)
       DRS = DSQRT( RSX*RSX + RSY*RSY + RSZ*RSZ )
C
C  Calculate the vector dot products.
```

```
      TSDOTV = ( TSX*VSX + TSY*VSY + TSZ*VSZ ) / DTS
      RSDOTV = ( RSX*VSX + RSY*VSY + RSZ*VSZ ) / DRS
C
C  Calculate the frequency at the satellite and at the receiver.
C
      FREQ1 = FREQ  * ( 1. - TSDOTV / VLIGHT )
      FREQ2 = FREQ1 * ( 1. - RSDOTV / VLIGHT )
C
C  Calculate the Doppler frequency.
C
      DOPLR = FREQ2 - FREQ
      RETURN
      END
```

```
                  'OUT     )OPC     [STA     AN,1
C
C *************************************************************************
C *                                                                       *
C *                          SUBROUTINE DOPCALC                           *
C *                                                                       *
C *************************************************************************
C
C  AUTHOR:       Dr. E. James Wadiak
C  DATE:         3-FEB-1988
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ.SIM]DOPCALC.FOR
C
C  CALLING ROUTINE:      MULTLSQ.FOR
C                        DOPDERIV.FOR
C
C  SUBROUTINES CALLED:  DOP.FOR
C
C  COMPILE INSTRUCTIONS:        via INCLUDE statement in MULTLSQ.FOR
C
C  LINK/LOAD INSTRUCTIONS:      via INCLUDE statement in MULTLSQ.FOR
C
C  PARENT PROGRAM:      MULTLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C     This subroutine calculates the Doppler frequency and chirp, both at time
C  TIME.  The actual Doppler frequency is calculated in subroutine DOP.  The
C  chirp (dDOPPLER/dt) is calculated by a finite difference method using
C  multiple calls to DOP.
C
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. Calculate the Doppler shift at time (TIME - 1millisecond).
C
C    2. Calculate the Doppler shift at time (TIME + 1millisecond).
C
C    3. Calculate the chirp as delta(DOPPLER)/delta(time).
C
C    4. Calculate the Doppler shift at time TIME.
C
C    5. RETURN to the calling program.
C
C
C  INPUTS   EXPLICIT (via arguments to the CALL statement):
C
C               ISTA - receiver station ID number.
C               ITRAN - transmitter station ID number.
C               TIME - time at which Doppler and chirp are to be calculated.
C
C
C           IMPLICIT (via COMMON block):
C
C               DOPLR - calculated Doppler shift returned from subroutine DOP.
C
C
C  OUTPUTS  EXPLICIT:    NONE
C
C           IMPLICIT (via COMMON block):
C
C               DOPLR - Doppler shift at time TIME.
C               DOPRAT - chirp at time TIME.
C
C  MAJOR VARIABLES:
C
```

```
(               US -   _ple:    ft     .me    US.
C                DOPLR - calculated Doppler shift returned from subroutine DOP.
C                DOPRAT - chirp at time TIME.
C                DPLUS - Doppler shift at time TPLUS.
C                ISTA - receiver station ID number.
C                ITRAN - transmitter station ID number.
C                TIME - time at which Doppler and chirp are to be calculated.
C                TMINUS - (TIME - 1 millisecond).
C                TPLUS - (TIME + 1 millisecond).
C
C  MODIFIED:
C
C
C*****************************************************************************
      INCLUDE 'SPACE:[WADIAK.LSQ.SIM]MULTLSQ.CMN'
C
C  First get the Doppler rate at time TIME.  Use finite differences
C  to get dDOPPLER/dT.
C
      TMINUS = TIME - 0.001
      CALL DOP(ISTA,ITRAN,TMINUS)
      DMINUS = DOPLR
      TPLUS = TIME + 0.001
      CALL DOP(ISTA,ITRAN,TPLUS)
      DPLUS = DOPLR
      DOPRAT = (DPLUS - DMINUS) / (TPLUS - TMINUS)
C
C  Next, get the Doppler frequency at time TIME.
C
      CALL DOP(ISTA,ITRAN,TIME)
      RETURN
      END
```

```
C
C *********************************************************************
C *                                                                   *
C *                        SUBROUTINE DOPDERIV                        *
C *                                                                   *
C *********************************************************************
C
C  AUTHOR:       Dr. E. James Wadiak
C  DATE:         3-FEB-1988
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ.SIM]DOPDERIV.FOR
C
C  CALLING ROUTINE:      MULTLSQ.FOR
C
C  SUBROUTINES CALLED:   DOPCALC.FOR
C
C  COMPILE INSTRUCTIONS:         via INCLUDE statement in MULTLSQ.FOR.
C
C  LINK/LOAD INSTRUCTIONS:       via INCLUDE statement in MULTLSQ.FOR.
C
C  PARENT PROGRAM:       MULTLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C     This subroutine calculates the partial derivatives of the Doppler and
C  Doppler rate (chirp), taken with respect to each of the parameters being
C  varied, at time TIME.  The partials are returned to the main program via
C  the COMMON block.
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. DO, for each of the parameters being varied,
C
C       1a. Store the current value of the parameter in a holding variable B.
C
C       1b. Set the current parameter equal to its original value plus a small
C           increment PARINC.
C
C       1c. Calculate the Doppler frequency and chirp.
C
C       1d. Set the current parameter equal to its original value minus a small
C           increment PARINC.
C
C       1e. Calculate the Doppler frequeny and chirp.
C
C       1f. Set the Doppler partial derivative equal to the difference between
C           the two Doppler measurements, divided by twice the increment PARINC.
C
C       1g. Set the chirp partial derivative equal to the difference between the
C           two chirp measurements, divided by twice the increment PARINC.
C
C       1h. Restore the current parameter to its original value on entry to this
C           subroutine.
C
C    2. ENDDO.
C
C    3. RETURN to the calling program.
C
C  INPUTS    EXPLICIT (via arguments to CALL statement):
C
C                 ISTA - receiver station ID number.
C                 ITRAN - transmitter station ID number.
C
C            IMPLICIT (via COMMON block):
C
```

```
               .R -   :ula   valu   f tl   ·ppl   iift   m DC   C.
C              DOPRAT - calculated values of the chirp from DOPCALC.
C              IVP(1-16) - array containing the ID numbers of the parameters
C                    being varied.
C              NVP - number of parameters being varied.
C              PAR(1-16) - current values of the parameters being varied.
C
C  OUTPUTS  EXPLICIT:   NONE
C
C           IMPLICIT (via COMMON block):
C
C              DPARTL(ISTA,ITRAN,J) - partial derivative of the Doppler shift
C                                     of the signal received at receiver ISTA
C                                     from transmitter ITRAN, taken with
C                                     respect to the Jth parameter.
C              RPARTL(ISTA,ITRAN,J) - partial derivative of the chirp of the
C                                     signal received at receiver ISTA from
C                                     transmitter ITRAN, taken with respect to
C                                     the Jth parameter.
C
C
C  MAJOR VARIABLES:
C
C              B - variable holding the value of the Jth parameter on entry to
C                  this subroutine.
C              DHIVAL - value of the Doppler shift associated with the
C                       incrementally increased value of the current parameter.
C              DLOVAL - value of the Doppler shift associated with the
C                       incrementally decreased value of the current parameter.
C              DOPLR - calculated values of the Doppler shift from DOPCALC.
C              DOPRAT - calculated values of the chirp from DOPCALC.
C              DPARTL(ISTA,ITRAN,J) - partial derivative of the Doppler shift
C                                     of the signal received at receiver ISTA
C                                     from transmitter ITRAN, taken with
C                                     respect to the Jth parameter.
C              ISTA - receiver station ID number.
C              ITRAN - transmitter station ID number.
C              IVP(1-16) - array containing the ID numbers of the parameters
C                    being varied.
C              NVP - number of parameters being varied.
C              PAR(1-16) - current values of the parameters being varied.
C              PARINC - incremental change to the parameters being varied.  Set
C                       in the subroutine to 1 meter and 1meter/second.
C              RHIVAL - value of the chirp associated with the incrementally
C                       increased value of the current parameter.
C              RLOVAL - value of the chirp associated with the incrementally
C                       decreased value of the current parameter.
C              RPARTL(ISTA,ITRAN,J) - partial derivative of the chirp of the
C                                     signal received at receiver ISTA from
C                                     transmitter ITRAN, taken with respect to
C                                     the Jth parameter.
C
C  MODIFIED:
C
C
C*********************************************************************
        INCLUDE 'SPACE:[WADIAK.LSQ.SIM]MULTLSQ.CMN/LIST'
C
C  Calculate the partials using finite differences.
C
        DO I=1,NVP
          J = IVP(I)
          B = PAR(J)
          PARINC = 1.0
          PAR(J) = B + PARINC
          CALL DOPCALC(ISTA,ITRAN,TIME)
```

```
  ..IVAL    JOPL..
  RHIVAL = DOPRAT
  PAR(J) = B - PARINC
  CALL DOPCALC(ISTA,ITRAN,TIME)
  DLOVAL = DOPLR
  RLOVAL = DOPRAT
  DPARTL(ISTA,ITRAN,J) = ( DHIVAL - DLOVAL ) / ( 2. * PARINC )
  RPARTL(ISTA,ITRAN,J) = ( RHIVAL - RLOVAL ) / ( 2. * PARINC )
  PAR(J) = B
ENDDO
RETURN
END
```

```
C
C ***************************************************************************
C *                                                                         *
C *                           SUBROUTINE MATPR                              *
C *                                                                         *
C ***************************************************************************
C
C  AUTHORS:      Dr. Michael D. Andrews, Dr. E. James Wadiak
C  DATE:         23-NOV-1987
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ.SIM]MATPR.FOR
C
C  CALLING ROUTINE:              MULTLSQ.FOR
C
C  SUBROUTINES CALLED:           NONE
C
C  COMPILE INSTRUCTIONS:         compiled via INCLUDE statement in MULTLSQ
C
C  LINK/LOAD INSTRUCTIONS:       linked via INCLUDE statement in MULTLSQ
C
C  PARENT PROGRAM:               MULTLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C   This subroutine prints out a lower triangular matrix which has been
C   previously stored in a vector array using subroutine IEL.
C
C  INPUTS   EXPLICIT (arguments to CALL statement):
C
C               A - vector array in which the matrix elements are stored.
C               LU - FORTRAN unit number to which output is directed.
C               M - dimension of the lower triangular matrix.
C               N - flag specifying which format to use on output.
C
C           IMPLICIT:   NONE
C
C  OUTPUTS  EXPLICIT:   matrix stored in array A is printed to output unit.
C
C           IMPLICIT:   NONE
C
C  MODIFIED:
C
C
C*****************************************************************************
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION A(78)
        DO  I=1,M
          IL=IEL(I,I)
          IG=IEL(I,1)
          IF(N.NE.1) THEN
            WRITE(LU,100) (A(J),J=IG,IL)
 100        FORMAT(5(2X,E11.3),/)
          ELSE
            WRITE(LU,101) (A(J),J=IG,IL)
 101        FORMAT(5(2X,F7.4))
          ENDIF
        ENDDO
        RETURN
        END
```

```
      SUBROUTINE MCALC(...,J)
C
C ***********************************************************************
C *                                                                     *
C *                         SUBROUTINE MCALC                            *
C *                                                                     *
C ***********************************************************************
C
C  AUTHOR:       Dr. Michael D. Andrews, Dr. E. James Wadiak
C  DATE:         3-FEB-1988
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ.SIM]MCALC.FOR
C
C  CALLING ROUTINE:      MULTLSQ.FOR
C                        MDERIV.FOR
C
C  SUBROUTINES CALLED:  NONE
C
C  COMPILE INSTRUCTIONS:         via INCLUDE statement in MULTLSQ.FOR
C
C  LINK/LOAD INSTRUCTIONS:       via INCLUDE statement in MULTLSQ.FOR
C
C  PARENT PROGRAM:      MULTLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C     This program will calculate the theoretical value of the phase difference
C  between the reference antenna and the JNth antenna at the current receiver
C  site, based on the satellite's current position.  All calculations are done
C  in the NAVSPASUR great circle coordinate system.  The receiving station
C  antenna positions, the current estimates of the satellite position and
C  velocity at TPRED (time of fence passage), and the time of the IIth data line
C  are all passed in the COMMON block.  The satellite position at the time of
C  the IIth data line is calculated as X + VT, ignoring gravitational
C  acceleration over the short time period during which the satellite is
C  illuminated by the transmitter beam.
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. Get the great circle coordinates of the JNth antenna via  COMMON block.
C
C
C    2. Get the great circle coordinates of the reference antenna via the
C       COMMON block.
C
C    3. Obtain the current estimates of the satellite position and velocity at
C       time TPRED via the COMMON block.  Use these to calculate the satellite
C       position and velocity at the time of the IIth data line.
C
C    4. Calculate the distance from the satellite to the JNth antenna.
C
C    5. Calculate the light travel time corrections from the satellite to the
C       receiver site. TURN OFF FOR SIMULATED DATA, FOR WHICH ALL CALCULATIONS
C       ARE IN SATELLITE TIME !
C
C    6. Calculate the distance from the satellite to the reference antenna.
C
C    7. Calculate the difference in path lengths from the satellite to the JNth
C       antenna and the reference antenna, respectively.  Express the result in
C       wavelengths at the Doppler-shifted receive frequency.
C
C    8. RETURN the result, CLC, to the calling program via the COMMON block.
C
C
C  INPUTS  EXPLICIT (via arguments to CALL statement):
C
```

```
             da     .ne      er.
C           J - baseline number.
C
C
C         IMPLICIT (via COMMON block):
C
C             FQREC(II) - receive frequency associated with IIth data line.
C             NREF(I) - # of the reference antenna at the Ith receiver site.
C             PAR(1-16) - current values of the parameters being varied.
C             POS(I,JN,1-3) - x,y,z coordinates of the JNth antenna at the Ith
C                             receiver site.
C             TOBS(II) - time of the IIth data line with respect to TPRED.
C
C                 (via INCLUDE CONSTANTS.FOR statement):
C
C             VLIGHT - velocity of light in meters/second.
C
C
C OUTPUTS  EXPLICIT:   NONE
C
C          IMPLICIT (via COMMON block):
C
C             CLC - calculated value of the ideal phase difference, in
C                   rotations.
C
C
C MAJOR VARIABLES:
C
C             CLC - calculated (ideal) phase difference, in rotations.
C             D - distance from the satellite to the JNth antenna.
C             DR - distance from satellite to the reference antenna.
C
C             DRX:
C             DRY: x,y,z distance from the satellite to the reference antenna.
C             DRZ:
C
C             DX:
C             DY: x,y,z distance from thesatellite to the JNth antenna.
C             DZ:
C
C             FQREC(II) - receive frequency associated with IIth data line.
C             II - data line number.
C             J - baseline number.
C             JN - antenna number.
C             NREF(I) - # of the reference antenna at the Ith receiver site.
C             PAR(1-16) - current values of the parameters being varied.
C             PATH - path length difference from satellite to reference
C                    antenna and JNth antenna, respectively (in meters).
C             POS(I,JN,1-3) - x,y,z coordinates of the JNth antenna at the Ith
C                             receiver site.
C
C             RX:
C             RY: NAVSPASUR great circle coordinates of the reference antenna.
C             RZ:
C
C             SATX:
C             SATY: position of the satellite at the time of the current data
C             SATZ: line.
C
C             TCORR - satellite-receiver light travel time correction.
C             TOBS(II) - time of the IIth data line with respect to TPRED.
C             VLIGHT - velocity of light in meters/second.
C             WVLNGH - received signal wavelength, in meters.
C
C             X:
C             Y: NAVSPASUR great circle coordinates of the JNth antenna
```

```fortran
C
C
C  MODIFIED:
C
C
C***********************************************************************
        INCLUDE 'SPACE:[WADIAK.LSQ.SIM]MULTLSQ.CMN/LIST'
        INCLUDE 'SPACE:[WADIAK.LSQ.SIM]CONSTANTS.FOR/LIST'
C
C  Get the positions of the JNth antenna and the reference antenna at
C  site IRX(II).  J is the baseline number (n-1 baselines per site).
C
        IF(J.LT.NREF(IRX(II))) THEN
            JN = J
          ELSE
            JN = J + 1
        ENDIF
        X = POS(IRX(II),JN,1)
        Y = POS(IRX(II),JN,2)
        Z = POS(IRX(II),JN,3)
        RX = POS(IRX(II),NREF(IRX(II)),1)
        RY = POS(IRX(II),NREF(IRX(II)),2)
        RZ = POS(IRX(II),NREF(IRX(II)),3)
C
C  Calculate the distance from the antenna to the satellite at time
C  TOBS(II).
C
C
C  NOTE:
C
C  The current version of SIMDAT/GNRTR calculates all times as satellite times;
C  i.e., no correction is made for light travel time to the different receiving
C  stations.  Therefore, light travel time corrections are disabled here.  When
C  necessary, one iteration is made to calculate TCORR, which is sufficient to
C  get the time correct to within about 1 part in 10**8.
C
C       T = TOBS(II) - PAR(IRX(II)+6)
C       SATX = PAR(1) + T * PAR(4)
C       SATY = PAR(2) + T * PAR(5)
C       SATZ = PAR(3) + T * PAR(6)
C       DX = SATX - X
C       DY = SATY - Y
C       DZ = SATZ - Z
C       D = DSQRT( DX*DX + DY*DY + DZ*DZ )
C       TCORR = D / VLIGHT
        TCORR = 0.
        T = TOBS(II) - PAR(IRX(II)+6) - TCORR
        SATX = PAR(1) + T * PAR(4)
        SATY = PAR(2) + T * PAR(5)
        SATZ = PAR(3) + T * PAR(6)
        DX = SATX - X
        DY = SATY - Y
        DZ = SATZ - Z
        D = DSQRT( DX*DX + DY*DY + DZ*DZ )
C
C  Calculate the distance from the reference antenna to the satellite
C  at time TOBS(II).
C
        DRX = SATX - RX
        DRY = SATY - RY
        DRZ = SATZ - RZ
        DR = DSQRT( DRX*DRX + DRY*DRY + DRZ*DRZ )
C
C  The path difference is just DR - D.  Convert the path length
C  difference to rotations at the receiving frequency FQREC(II).
```

```
    tur    is \     to     cal   . pro.
C
        PATH =  DR - D
        WVLNGH = VLIGHT / FQREC(II)
        CLC = PATH / WVLNGH
        RETURN
        END
```

```
                    OUT     IDER     (,K)
C
C ***************************************************************************
C *                                                                         *
C *                         SUBROUTINE MDERIV                               *
C *                                                                         *
C ***************************************************************************
C
C  AUTHOR:       Dr. E. James Wadiak
C  DATE:         3-FEB-1988
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ.SIM]MDERIV.FOR
C
C  CALLING ROUTINE:      MULTLSQ.FOR
C
C  SUBROUTINES CALLED:  MCALC.FOR
C
C  COMPILE INSTRUCTIONS:         via INCLUDE statement in MULTLSQ.FOR
C
C  LINK/LOAD INSTRUCTIONS:       via INCLUDE statement in MULTLSQ.FOR
C
C  PARENT PROGRAM:       MULTLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C
C     This program uses a finite difference procedure to calculate the partial
C  derivatives of the phase difference for the Kth baseline at the current
C  receiver site.  On each call to this subroutine artial derivatives are
C  calculated with respect to each of the parameters being varied in the least
C  squares routine.
C
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. DO, for each parameter being varied in the least squares routine,
C
C       1a. Check to be sure that the parameter is not a clock offset at a
C           receiver site OTHER THAN the currenyt site.  If so, bypass to the
C           next parameter.  Otherwise,
C
C       1b. Save the current value of the parameter in a holding variable B.
C
C       1c. Set the parameter equal to its starting value plus a small increment
C           PARINC, and calculate the expected phase difference.
C
C       1d. Set the parameter equal to its starting value minus a small
C           increment PARINC, and calculate the expected phase difference.
C
C       1e. Set the partial derivative with respect to the current parameter
C           equal to the change in the expected phase differences divided by
C           twice PARINC.
C
C       1f. Reset the parameter to its value at the time of entry to the
C           subroutine.
C
C    2. ENDDO.
C
C    3. Return the values of all partials via the COMMON block, and RETURN to
C       the calling program.
C
C
C  INPUTS   EXPLICIT (via arguments to the CALL statement):
C
C                II - data line number.
C                K - baseline number.
```

```
C           IMPLICIT (via COMMON block):
C
C               CLC - calculated (ideal) phase difference value returned from
C                     subroutine MCALC.
C               IRX(II) - receiver number for the IIth data line.
C               IVP(I) - parameter ID number for the Ith varied parameter.
C               NVP - number of varied parameters.
C
C  OUTPUTS  EXPLICIT (via arguments to CALL MCALC statements):
C
C               II - data line number.
C               K - baseline number.
C
C           IMPLICIT (via COMMON block):
C
C               PARTL(1-16) - partial derivatives of the phase difference, taken
C                             with respect to each of the parameters being
C                             varied.
C
C  MAJOR VARIABLES:
C
C               B - dummy variable to hold the value of the parameter on entry
C                   to this subroutine.  Used to return the parameter to its
C                   entry value prior to exiting.
C               CLC - calculated (ideal) phase difference value returned from
C                     subroutine MCALC.
                II - data line number.
C               IRX(II) - receiver number for the IIth data line.
C               IVP(I) - parameter ID number for the Ith varied parameter.
C               K - baseline number.
C               NUM - parameter ID# of the clock offset at the current receiver.
C               NVP - number of varied parameters.
C               PARINC - amount to increment and decrement the parameter value
C                        by in order to get the finite difference partials.
C               PARTL(1-16) - partial derivatives of the phase difference, taken
C                             with respect to each of the parameters being
C                             varied.
C               PLS - variable to hold the expected phase difference associated
C                     with the incremented parameter value.
C
C
C  MODIFIED:
C
C
C******************************************************************************
        INCLUDE 'SPACE:[WADIAK.LSQ.SIM]MULTLSQ.CMN/LIST'
C
C   Set all the partials to zero.
C
        DO I=1,NVP
          PARTL(I) = 0.
        ENDDO
C
C   Calculate the partials.
C
        DO I=1,NVP
          J=IVP(I)
C
C  The partials for clock offsets at stations other than IRX(II)
C  are zero. Don't calculate these.
C
          NUM = IRX(II) + 6
          IF(J.LE.6.OR.J.EQ.NUM) THEN
            B = PAR(J)
            PARINC = 1.0
```

```
      PAR(     B       INC
      CALL MCALC(II,K)
      PLS=CLC
      PAR(J) = B - PARINC
      CALL MCALC(II,K)
      PARTL(J) = (PLS - CLC)/( 2. * PARINC )
      PAR(J) = B
    ENDIF
  ENDDO
  RETURN
  END
```

```
C
C ***********************************************************************
C *                                                                     *
C *                         SUBROUTINE NRMEQ                            *
C *                                                                     *
C ***********************************************************************
C
C  AUTHOR:       Dr. Michael D. Andrews, Dr. E. James Wadiak
C  DATE:         23-NOV-1987
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ.SIM]NRMEQ.FOR
C
C  CALLING ROUTINE:              MULTLSQ.FOR
C
C  SUBROUTINES CALLED:           NONE
C
C  USER-DEFINED
C  FUNCTIONS CALLED:             IEL (in IEL.FOR)
C
C  COMPILE INSTRUCTIONS:         compiled via INCLUDE statement in MULTLSQ
C
C  LINK/LOAD INSTRUCTIONS:       linked via INCLUDE statement in MULTLSQ
C
C  PARENT PROGRAM:               MULTLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C   This subroutine increments the normal equations of the nonlinear least
C   squares program MULTLSQ.  Each call to NRMEQ increments the normal
C   equations for one phase difference data residual.
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. Get the error associated with the datum through the COMMON block.
C       Set the data weight GG equal to 1./(error)**2.
C
C    2. DO, for each varied parameter (i.e., NVP normal equations),
C
C       2a. Increment the right-hand-side by the weighted data residual
C           times the partial derivative of the phase difference for the
C           current receiver and antenna, taken with respect to the current
C           parameter.
C
C       2b. DO, for each lower triangular matrix element on the left-
C           hand-side of the normal equations,
C
C           2b(1). Increment the left-hand side by the weighted product
C                  of the partial derivatives of the phase difference
C                  associated with the respective row and column numbers.
C
C       2c. END of both loops.
C
C    3. RETURN to the main program.
C
C  INPUTS   EXPLICIT (arguments to CALL statement):
C
C                IOBS - number of the data residual being processed.
C
C           IMPLICIT (via COMMON block):
C
C                ALHS(1-16) - current left-hand-side of the normal equations.
C                DELTA - RMS error associated with the IOBSth data residual.
C                IVP(1-6) - array containing the parameter number of each of
C                           the NVP parameters being varied.
C                NVP - number of parameters being varied in least squares fit.
```

```
              'L(1    - pa    l d     tiv    th   ase    erer    ake
C                      with respect to each of the varied parameters.
C               RHS(1-6) - current right-hand-side of the normal equations.
C
C  OUTPUTS  EXPLICIT: NONE
C
C           IMPLICIT (via COMMON block):
C
C               ALHS(1-16) - incremented left-hand-side of the normal equations.
C               RHS(1-6) - incremented right-hand-side of the normal equations.
C
C  MAJOR VARIABLES:
C
C               ALHS(1-16) - current left-hand-side of the normal equations.
C               DELTA - RMS error associated with the IOBSth data residual.
C               IOBS - ID number of the data residual being processed.
C               IVP(1-6) - array containing the parameter number of each of
C                     the NVP parameters being varied.
C               NVP - number of parameters being varied in least squares fit.
C               PARTL(1-6) - partial derivatives of the phase difference
C                      with respect to each of the varied parameters.
C               RHS(1-6) - current right-hand-side of the normal equations.

C
C  MODIFIED:
C
C
C*****************************************************************************
      INCLUDE 'SPACE:[WADIAK.LSQ]MULTLSQ.CMN/LIST'
      GG=1. / ( DELTA * DELTA )
      DO I=1,NVP
        IA = IVP(I)
        RHS(I) = RHS(I) + GG * RESID(IOBS) * PARTL(IA)
        DO J=1,I
          K = IEL(I,J)
          IB = IVP(J)
          ALHS(K) = ALHS(K) + GG * PARTL(IA) * PARTL(IB)
        ENDDO
      ENDDO
      RETURN
      END
```

```
      SUBROUTINE RANDG(IX,IY,SIGMA,RMEAN,VAL)
C
C ***********************************************************************
C *                                                                     *
C *                         SUBROUTINE RANDG                            *
C *                                                                     *
C ***********************************************************************
C
C AUTHOR:        Dr. E. James Wadiak
C DATE:          26-JAN-1988
C LANGUAGE:      FORTRAN ANSI-77  (VAX/VMS operating system)
C FILE:          VX7770::SPACE:[WADIAK.LSQ.SIM]RANDG.FOR
C
C CALLING ROUTINES:              ERROR.FOR
C                                MULTLSQ.FOR
C
C SUBROUTINES CALLED:            RANDU - VAX library subroutine which returns
C                                        a random number uniformly distributed
C                                        between 0 and 1.
C
C COMPILE INSTRUCTIONS:          via INCLUDE statement in parent program.
C
C LINK/LOAD INSTRUCTIONS:        via INCLUDE statement in parent program.
C
C PARENT PROGRAMS:               SIMDAT.FOR, MULTLSQ.FOR
C
C PROGRAM DESCRIPTION:
C
C    This subroutine applies the Central Limit Theorem to derivea random
C number VAL whose distribution is Gaussian with a characteristic dispersion
C of SIGMA and a mean value of RMEAN.
C
C
C PROGRAM ALGORITHM (PSEUDOCODE):
C
C   1. Sum 12 random numbers uniformly distributed between 0 and 1.  The
C      resultant number is Gaussian-distributed about the expection value of
C      <6> and a standard deviation pf 1.
C
C   2. Multiply the deviation from the expectation value times the desired
C      standard deviation, and add the desired mean.  This produces a random
C      number with the desired properties.
C
C   3. RETURN to the calling program.
C
C INPUTS   EXPLICIT (via arguments to the CALL statement):
C
C                IX - random number generator seed.
C                IY - random number generator seed.
C                RMEAN - desired mean of the output random number.
C                SIGMA - desired standard deviation of the output random number.
C
C
C          IMPLICIT:   NONE
C
C OUTPUTS  EXPLICIT (via the arguments to the CALL statement):
C
C                IX - new seed for next call to RANDG.
C                IY - new seed for next call to RANDG.
C                VAL - random number with the desired distribution properties.
C
C
C          IMPLICIT:   NONE
C
C MAJOR VARIABLES:
C
```

```
C                  sum    he    .ifo    dis    ute‹    dom    ›ers
C                RMEAN - desired mean of output random number.
C                SIGMA - desired standard deviation of output random number.
C                VAL - output random number with desired properties.
C
C
C  MODIFIED:
C
C
C***********************************************************************
        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*4 Y
        A = 0.0
        DO I=1,12
          CALL RANDU(IX,IY,Y)
          A = A + Y
        ENDDO
        VAL = ( A - 6.0 ) * SIGMA + RMEAN
        RETURN
        END
```

```
      SUBROUTINE RNRMEQ(ID,IOBS)
C
C **************************************************************************
C *                                                                        *
C *                          SUBROUTINE RNRMEQ                             *
C *                                                                        *
C **************************************************************************
C
C  AUTHOR:       Dr. E. James Wadiak
C  DATE:         25-FEB-1988
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ.SIM]RNRMEQ.FOR
C
C  CALLING ROUTINE:      MULTLSQ.FOR
C
C  SUBROUTINES CALLED:  NONE
C
C  USER-DEFINED          IEL.FOR
C  FUNCTIONS CALLED:
C
C  COMPILE INSTRUCTIONS:         via INCLUDE statement in MULTLSQ.FOR
C
C  LINK/LOAD INSTRUCTIONS:       via INCLUDE statement in MULTLSQ.FOR
C
C  PARENT PROGRAM:       MULTLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C     This subroutine increments the normal equations of the nonlinear least
C  squares fitting program MULTLSQ.  Each call to DNRMEQ increments the normal
C  equations for one Doppler rate (chirp) datum residual.
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. Get the error associated with the datum via the COMMON block.  Set the
C       data weight equal to 1./(error)**2.
C
C    2. DO, for each varied parameter (i.e., NVP normal equations).
C
C       2a. Increment the right-hand-side of the Ith normal equation by the
C           weighted data residual times the partial derivative of the chirp
C           taken with respect to the Ith parameter.
C
C       2b. DO, for each lower triangular matrix element on the left-hand-side
C           of the Ith normal equation,
C
C           2b(1). Increment the left-hand-side by the weighted product of the
C                  partial derivatives of the chirp taken with respect to the
C                  Ith (row) and Jth (column) parameters.
C
C    3. END both DO loops.
C
C    4. RETURN to the calling program.
C
C  INPUTS    EXPLICIT (via arguments to the CALL statement):
C
C                ID - array index specifying which chirp datum to use.
C                IOBS - array index specifying which residual datum to use.
C
C            IMPLICIT (via COMMON block):
C
C                ALHS(1-16) - current left-hand-sides of the normal equations.
C                DELTA - RMS error associated with the IOBSth data residual.
C                IVP(1-6) - array containing the parameter ID numbers of each of
C                           the NVP parameters being varied.
C                NVP - number of parameters being varied in least squares fit.
```

```
C               RHS(1-16) - current right-hand-sides of the normal equations.
C               RPARTL(IR,IT,I) - partial derivative of the chirp of the signal
C                                 from transmitter IT received at station IR,
C                                 taken with respect to the Ith parameter.
C
C
C  OUTPUTS  EXPLICIT:   NONE
C
C           IMPLICIT (via COMMON block):
C
C               ALHS(1-16) - incremented left-hand-sides of normal equations.
C               RHS(1-16) - incremented right-hand-sides of normal equations.
C
C
C  MAJOR VARIABLES:
C
C               ALHS(1-16) - left-hand-sides of the normal equations.
C               DELTA - RMS error associated with the IOBSth residual.
C               RPARTL(IR,IT,I) - partial derivative of the chirp of the signal
C                                 from transmitter IT received at station IR,
C                                 taken with respect to the Ith parameter.
C               GG - weight assigned to the current datum.
C
C
C  MODIFIED:
C
C
C**********************************************************************
      INCLUDE 'SPACE:[WADIAK.LSQ.SIM]MULTLSQ.CMN/LIST'
      GG = 1. / ( DELTA * DELTA )
      DO I=1,NVP
        IA = IVP(I)
        PARTLA = RPARTL(KDOPR(ID),KDOPT(ID),IA)
        RHS(I) = RHS(I) + GG * RESID(IOBS) * PARTLA
        DO J=1,I
          K = IEL(I,J)
          IB = IVP(J)
          PARTLB = RPARTL(KDOPR(ID),KDOPT(ID),IB)
          ALHS(K) = ALHS(K) + GG * PARTLA * PARTLB
        ENDDO
      ENDDO
      RETURN
      END
```

```
C
C ***********************************************************************
C *                                                                     *
C *                        SUBROUTINE SYMIN                             *
C *                                                                     *
C ***********************************************************************
C
C  AUTHOR:       Unknown
C  DATE:         09-JUL-1987 (last modification)
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ.SIM]SYMIN.FOR
C
C  CALLING ROUTINE:              MULTLSQ.FOR
C
C  SUBROUTINES CALLED:           NONE
C
C  COMPILE INSTRUCTIONS:         compiled via INCLUDE statement in MULTLSQ
C
C  LINK/LOAD INSTRUCTIONS:       linked via INCLUDE statement in MULTLSQ
C
C  PARENT PROGRAM:               MULTLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C   This subroutine is a working matrix inversion algorithm which dates
C   back to the early days of electronic computers.  We plan to replace
C   this program element someday with a modernized version.  Extensive
C   documentation of this element is therefore unwarranted, and it is
C   submitted as is.
C
C************************************************************************
        INCLUDE 'SPACE:[WADIAK.LSQ]MULTLSQ.CMN/LIST'
      DIMENSION PVCOL(27), PVROW(27),IUSE(27)
C
C     INITIALLY ALL ROWS ARE UNUSED
C
      DO 10 I=1,NVP
10    IUSE(I)=0
      DO 130 ITER=1,NVP
C     SEARCH FOR THE LARGEST DIAGONAL ELEMENT FOR USE AS A PIVOT
      DMAX=0.D0
      II=1
      DO 40 I=1,NVP
      DIAG=DABS(ALHS(II))
      IF(DIAG-DMAX) 40,40,20
20    IF(IUSE(I)) 30,30,40
30    DMAX=DIAG
      K=I
      PIVOT=ALHS(II)
40    II=II+I+1
      IF(DMAX)200,200,50
 50   IUSE(K)=1
C     GET THE ELEMENTS OF THE PIVOT ROW AND COLUMN
C     WITH THE PROPER SIGN
      IK=1+(K*(K-1))/2
      DO 120 I=1,NVP
      IF(I-K) 60,80,90
   60 CONTINUE
      PVROW(I)=ALHS(IK)
      PVCOL(I)=ALHS(IK)/PIVOT
      ALHS(IK)=0.D0
      IK=IK+1
      IF(IUSE(I)) 120,120,70
70    PVCOL(I)=-PVCOL(I)
      GO TO 120
```

```
80     PVROW(I)=1.D0
       PVCOL(I)=-1.D0/PIVOT
       ALHS(IK)=0.D0
       PVRWB=RHS(I)
       RHS(I)=0.D0
       GO TO 110
    90 CONTINUE
       PVROW(I)=ALHS(IK)
       PVCOL(I)=ALHS(IK)/PIVOT
       ALHS(IK)=0.D0
       IF(IUSE(I)) 110,110,100
100    PVROW(I)=-PVROW(I)
   110 IK=IK+I
120    CONTINUE
C      PERFORM THE PIVOT STEP
       IJ=1
       DO 130 I=1,NVP
       RHS(I)=RHS(I)-PVCOL(I)*PVRWB
       DO 130 J=1,I
       ALHS(IJ)=ALHS(IJ)-PVCOL(I)*PVROW(J)
130    IJ=IJ+1
       RETURN
200    WRITE(LU,210) ITER,N, (IUSE(I),I=1,N)
   210 FORMAT(' FAILURE TO FIND NON-ZERO DIAGONAL ELEMENT IN SYMIN ON
      .ITERATION  ',I3,' FOR N= ',I3,' IUSE = ',/,(1X,50I2))
       RETURN
       END
```

```
                        .*8     TION    (INT
C
C *********************************************************************
C *                                                                   *
C*                          FUNCTION ERR                              *
C *                                                                   *
C *********************************************************************
C
C  AUTHORS:      Dr. Michael D. Andrews, Dr. E. James Wadiak
C  DATE:         23-NOV-1987
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ.SIM]ERR.FOR
C
C  CALLING ROUTINE:              MULTLSQ.FOR
C
C  SUBROUTINES CALLED:           NONE
C
C  COMPILE INSTRUCTIONS:         compiled via INCLUDE statement in MULTLSQ
C
C  LINK/LOAD INSTRUCTIONS:       linked via INCLUDE statement in MULTLSQ
C
C  PARENT PROGRAM:               MULTLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C      This function calculates the expected RMS error on a phase difference
C   datum based on the received signal strength of the datum.  The RMS errors
C   have been modelled as a fourth order polynomial in dB above the noise
C   floor.
C
C     The current error model is based on the four-hour NAVSPASUR data
C  tape T5321.  The model was generated from the combined data for
C  all stations except station #5, with N-S and E-W baselines combined.
C
C
C  INPUTS   EXPLICIT (arguments to FUNCTION statement):
C
C                INT - amplitude of the received signal.
C
C           IMPLICIT:   A0 - A4: coefficients to the fourth order polynomial
C                                fit to the error model.  These are set by
C                                assignment statements in the code.
C
C  OUTPUTS  EXPLICIT:   NONE
C
C           IMPLICIT (returned as FUNCTION value):
C
C                ERR(INT) - RMS error associated with the data point amplitude.
C
C  MAJOR VARIABLES:
C
C                NDB - signal strength in dB above assumed -152dBm noise floor.
C                A0 - zeroth order coefficient of polynomial.
C                A1 - first order coefficient of polynomial.
C                A2 - second order coefficient of polynomial.
C                A3 - third order coefficient of polynomial.
C                A4 - fourth order coefficient of polynomial.
C
C  MODIFIED:
C
C
C**********************************************************************
        IMPLICIT REAL*8 (A-H,O-Z)
        NDB = 152 - INT
        A0 =  1.31D-1
        A1 = -1.23D-2
```

```
     5.      4
A3 = -4.48D-6
A4 = -1.02D-7
ERR = A4*NDB**4 + A3*NDB**3 + A2*NDB**2 + A1*NDB + A0
RETURN
END
```

```
                            F'     ON      :,J)
C
C ***********************************************************************
C *                                                                     *
C *                          FUNCTION IEL                               *
C *                                                                     *
C ***********************************************************************
C
C  AUTHOR:       Dr. Michael D. Andrews
C  DATE:         23-NOV-1987
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ.SIM]IEL.FOR
C
C  CALLING ROUTINES:             MULTLSQ.FOR
C                                NRMEQ.FOR
C                                DNRMEQ.FOR
C                                RNRMEQ.FOR
C
C  SUBROUTINES CALLED:           NONE
C
C  COMPILE INSTRUCTIONS:         compiled via INCLUDE statement in MULTLSQ
C
C  LINK/LOAD INSTRUCTIONS:       linked via INCLUDE statement in MULTLSQ
C
C  PARENT PROGRAM:               MULTLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C   This function is a mapping routine designed to allow storage of
C   an inherently 2-dimensional matrix into a 1-dimensional vector.
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C  INPUTS   EXPLICIT (arguments to FUNCTION statement):
C
C               I - row number of the matrix element.
C               J - column number of the matrix element.
C
C           IMPLICIT:   NONE
C
C  OUTPUTS  EXPLICIT:   NONE
C
C           IMPLICIT (returned as FUNCTION value):
C
C               IEL(I,J) - vector index corresponding to matrix element (I,J)
C
C  MODIFIED:
C
C
C************************************************************************
      IF(J.LE.I) THEN
         K = I
         L = J
        ELSE
         K = J
         L = I
      ENDIF
      IEL = (( K*K - K ) / 2 ) + L
      RETURN
      END
```